

**VŠB – Technická univerzita Ostrava**

**Fakulta elektrotechniky a informatiky**

**Katedra kybernetiky a biomedicínského inženýrství**

**Analýza polovodičových pamětí pod vlivem  
ionizujícího záření v reálném čase**

**Real Time Analysis of Semiconductor  
Memories under the Influence of Ionizing  
Radiation**

## Zadání diplomové práce

Student:

**Bc. Zuzana Chvostková**

Studijní program:

N2649 Elektrotechnika

Studijní obor:

3901T009 Biomedicínské inženýrství

Téma:

**Analýza polovodičových pamětí pod vlivem ionizujícího záření  
v reálném čase  
Real Time Analysis of Semiconductor Memories under the Influence  
of Ionizing Radiation**

Zásady pro vypracování:

1. Seznámení se s technologií programovatelných logických obvodů FPGA a vlastnostmi ionizujícího záření používaného v biomedicině.
2. Návrh koncepce přípravku pro průběžnou analýzu obsahu polovodičových pamětí.
3. Návrh obvodové logiky FPGA přípravku pro průběžnou analýzu vybraných typů pamětí SRAM a EPROM.
4. Návrh a realizace DPS pro navržený přípravek.
5. Uskutečnění měření obsahu pamětí pod vlivem ionizujícího záření.
6. Zhodnocení dosažených výsledků.

Seznam doporučené odborné literatury:

- [1] ŠTASTNÝ, Jakub. *FPGA prakticky*. 1.vyd. Praha: BEN-technická literatura, 2011. 200 s. ISBN 978-80-7300-261-9.
- [2] HEINKEL, Ulrich et. al. *The VHDL Reference*. Chichester: John Wiley, 2000. ISBN 978-0-471-89972-3.
- [3] ULLMANN, Vojtěch. *Aplikace ionizujícího záření v radiologických oborech*. Ostrava : Ostravská univerzita, 2013. 182 s. ISBN 978-80-7464-211-1.
- [4] ULLMANN, Vojtěch. *Jaderná a radiační fyzika*. Ostrava : Ostravská univerzita, 2013. 173 s. ISBN 978-80-7368-669-7.
- [5] VALÁŠEK, Pavel a Roman LOSKOT. *Polovodičové paměti*. Praha : BEN - technická literatura, 2002. 224 s. ISBN 80-86056-79-1.



Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Ing. Vladimír Kašík, Ph.D.**

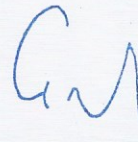
Datum zadání: 01.09.2013

Datum odevzdání: 07.05.2015



---

doc. Ing. Jiří Koziorek, Ph.D.  
*vedoucí katedry*



---

prof. RNDr. Václav Snášel, CSc.  
*děkan fakulty*

## **Abstrakt**

Cílem diplomové práce je zkoumat a následně potvrdit nebo vyvrátit vliv ionizujícího záření na různé typy polovodičových pamětí. Stručně je rozebrána problematika ionizujícího záření a technologie FPGA. Praktickou část práce tvoří návrh přípravku pro průběžnou analýzu obsahu polovodičových pamětí a následná realizace DPS pro navržený přípravek. Průběžná analýza obsahu paměti je během ozařování provedena pomocí obvodové logiky FPGA.

## **Abstract**

The objective of this thesis is to research and then prove or disprove the influence of ionising radiation on various types of semiconductor memories. The issue of ionising radiation and FPGA technique is briefly described in theoretical part of thesis. The practical part is dedicated to design of hardware component for continuous analysis of memories content and following realization of PCB of designed component. The continuous analysis of memory content during radiation is realized by FPGA circuit logic.

## **Klíčová slova**

Ionizující záření, FPGA, polovodičové paměti, SRAM, EPROM, EEPROM, stavový automat

## **Key words**

Ionizing radiation, FPGA, semiconductor memories, SRAM, EPROM, EEPROM, finite state machine

## Prohlášení

„Prohlašuji, že jsem tuto diplomovou práci vypracovala sama. Uvedla jsem všechny literární prameny a publikace, ze kterých jsem čerpala.“

4.5.2015

A handwritten signature in cursive script, reading "Zuzana Chvostková".

Zuzana Chvostková

## **Poděkování**

Ráda bych poděkovala vedoucímu diplomové práce panu Ing. Vladimíru Kašíkovi Ph.D. za cenné rady, velmi si vážím času, který mi věnoval během diskuzí a panu Ing. Lukáši Knybelovi za umožnění realizace měření ve Fakultní nemocnici Ostrava. Dále bych chtěla velmi poděkovat svým rodičům, kteří mě po celou dobu studia podporovali.

## Seznam zkratek

<b>CLB</b>	konfigurovatelný logický blok (configurable logic block)
<b>CMOS</b>	Complementary Metal Oxide Semiconductor
<b>DFS</b>	frekvenční syntéza
<b>DIP</b>	Dual-In-Line-package
<b>DLL</b>	závěs zpoždění (Delay – Locked Loop)
<b>DPS</b>	deska plošných spojů
<b>FSM</b>	Stavový automat (Finite State Machine)
<b>HD</b>	Hierarchický návrh (Hierarchical Design)
<b>IOB</b>	vstupně - výstupní blok (input/output block)
<b>LSB</b>	Least Significant Bit
<b>LUT</b>	Look Up Table
<b>MSB</b>	Most Significant Bit
<b>MSM</b>	Main State Machine
<b>MUX</b>	multiplexor
<b>SOIC</b>	Small-Outline-integrated-Circuit
<b>SPI</b>	sériové periferní rozhraní (Serial Peripheral Interface)
<b>SRAM</b>	statická paměť RAM
<b>UV</b>	ultrafialové záření
<b>X</b>	rentgenové záření (X záření)

# Obsah

1.	Úvod.....	1
2.	Vlastnosti ionizujícího záření používaného v biomedicině.....	2
2.1	Záření .....	2
2.2	Ionizující záření.....	3
3.	Výrobní technologie polovodičových pamětí .....	8
3.1.	Bipolární technologie.....	8
3.2.	Unipolární technologie.....	9
4.	Technologie FPGA.....	10
4.1.	Anatomie programovatelného obvodu .....	10
5.	Dosavadní poznatky ve vztahu ionizujícího záření a polovodičů .....	14
5.1.	Účinky záření na základní polovodičové součástky.....	14
6.	Návrh koncepce přípravku .....	16
6.1	Sledované paměťové obvody .....	17
6.2	Návrh přípravku .....	21
6.3	Návrh a realizace DPS pro navržený přípravek .....	25
7.	Vlastnosti FPGA řady Spartan6.....	29
7.1	CLB, Vrstvy a LUT.....	29
7.2	Bloková paměť RAM.....	30
7.3	CellularRAM.....	30
7.4	Digital Clock Manager (DCM) .....	32
8.	Návrh obvodové logiky v jazyce VHDL.....	33
8.1	DP2.....	34
8.2	main_state_machine (MSM).....	38



8.3	programovani .....	40
8.4	Programovani_EEPROM.....	46
8.5	SPI_Controller .....	50
8.6	LOG .....	51
8.7	Transmitter .....	58
8.8	BRAM.....	59
8.9	zakmit, detektor_hran.....	60
8.10	segDisplay .....	60
9.	Simulace a implementace návrhu, testování přípravku.....	61
9.1	Simulace programování SRAM .....	61
9.2	Simulace programování a čtení EEPROM.....	62
9.3	Simulace funkce „LOG“ .....	64
9.4	Implementace návrhu do FPGA.....	65
9.5	Testování přípravku .....	66
10.	Vizualizace výsledků .....	69
10.1	MemtestTerminal .....	69
10.2	GUI_DP .....	70
11.	Kontrolní měření .....	71
12.	Experimentální měření .....	72
12.1	Ozařovač TERAGAM.....	72
12.2	CyberKnife.....	73
12.3	Realizace měření .....	74
13.	Zhodnocení dosažených výsledků.....	80
14.	Použitá literatura .....	81

15.	Přílohy.....	83
-----	--------------	----

# 1.Úvod

Mnoho moderních elektronických systémů pracuje v prostředí, kde jsou stále vystaveny různým druhům záření, ať už z přirozených zdrojů nebo z člověkem uměle vyrobených zdrojů. Senzitivitu polovodičových komponent na ionizující záření dokazuje samotný fakt, že mnoho dnešních detektorů ionizujícího záření používá právě polovodičové elementy jako senzory. Neustále se zvyšující úroveň integrace elektronických komponent, požadavky na nízký výkon a miniaturizace mohou mít negativní dopad na komponenty senzitivní k ionizujícímu záření, jejich senzitivitu totiž zvyšují.

Paměť je důležitou součástí každého elektronického zařízení. Radiace může přímo nebo nepřímo způsobit lokální chyby uvnitř obvodu (poškození paměti obsahu) a porušit aktuální datové stavy. V důsledku tedy i změna jediného bitu paměti obsahu může znamenat pro správnou funkci součástky fatální následky a způsobit zaseknutí celého systému. Z toho hlediska představuje paměť nejslabší článek v celém řetězu systému. Jelikož dosud nebylo publikováno mnoho studií, které by se zabývaly nastíněným problémem, není ani dostatečně znám vliv na funkčnost této elektroniky. Motivací a cílem celé práce tedy je zkoumat a popsat vliv ionizujícího záření na vybrané druhy polovodičových pamětí.

V oblasti zdravotnictví se polovodičové komponenty – paměti používají například v kardiostimulátorech. Hlavní funkcí kardiostimulátoru je snímat, prostřednictvím vodičů připojených k srdečnímu svalu, elektrickou aktivitu srdce, a pokud je nutné elektricky srdce stimulovat. Senzitivita a funkce stimulatoru se nastavuje podle individuálních potřeb pacienta naprogramováním stimulatoru. Za selhání stimulatoru se považuje jak stimulace neodpovídající aktuálním potřebám pacienta, tak nevyzpytatelná či nevhodná stimulace. Nedostatečná stimulace je důsledkem selhání elektroniky. Nevyzpytatelná či nevhodná stimulace vzniká, pokud dojde ke špatnému naprogramování stimulatoru v důsledku lokálního poškození čipu. Jelikož tyto stavy jsou život ohrožující, je třeba znát možná rizika a předejít tak potenciálním problémům.

## 2. Vlastnosti ionizujícího záření používaného v biomedicině

### 2.1 Záření

Pojem záření (radiace) zahrnuje procesy, při nichž dochází k přenosu energie prostorem „na dálku“ prostřednictvím fyzikálních polí nebo mikročástic. „Kromě energie dochází při záření k přenosu hmoty a informace.“ [1]

**Přenos energie** je realizován dvěma druhy mechanismů:

- „**Časově proměnné pole**, které se šíří prostorem ve formě vln, jež se odpoutávají od zdroje a přenášejí do prostoru část energie z tohoto zdroje, např. elektromagnetické vlny.“ [1]
- „**Pohybující se částice**, které jsou emitovány zdrojem, velkou rychlostí letí prostorem, a přenášejí tak kinetickou energii (též hybnost, popř. elektrický náboj) ze zdroje do okolního prostoru, např. záření  $\beta$ , záření  $\alpha$ .“ [1]

**Šíření záření:**

- **Ve vakuu** dochází k volnému šíření vln a také k volnému pohybu částic (platí zákon setrvačnosti)
- **V látkovém prostředí** část záření projde původním směrem, část záření je rozptýlena či absorbována. „Míra rozptylu a absorpce je energeticky závislá, v důsledku čehož při průchodu záření látkou dochází nejen k zeslabení záření, ale často i ke změně spektrálního rozložení a ke vzniku dalšího sekundárního záření.“ [1]

**Dělení z hlediska energie kvant záření:**

- „**Měkké**“ **záření** se vyznačuje tak nízkou energií kvant ( $< \text{cca } 5\text{keV}$ ), že není schopno vyřážet elektrony z atomových obalů. Jeho účinky na látku jsou především mechanické a tepelné (popř. elektrické – fotoefekt, změny elektrické vodivosti; fotochemické – klasická fotografie). [1]
- „**Tvrdé**“ **záření** naopak disponuje vysokou energií kvant (desítky keV a vyšší), tudíž je schopno při průchodu látkou vyřážet elektrony z atomů a látku ionizovat. „Ionizace pak vede k nejen k účinkům elektrickým a fotochemickým, ale v případě sloučenin k řadě chemických reakcí rozkladu stávajících molekul a příp. vzniku sloučenin nových.“ [1] Těchto účinků se využívá v řadě radiačních technologií, v medicínské oblasti (např. léčení nádorových onemocnění.)

## 2.2 Ionizující záření

„Ionizujícím zářením nazýváme takové záření, jehož kvanta mají natolik vysokou energii, že jsou schopna vyřážet elektrony z atomového obalu a tím látku ionizovat.“ [1] Jak již bylo zmíněno v předchozím odstavci, za tuto energetickou hranici se pro běžné druhy záření (fotonové, elektronové a alfa) považuje 5keV.

### 2.1.1. Druhy ionizujícího záření

**Dělení z hlediska interakce záření s hmotou:**

- **Přímo ionizující** – kvanta záření nesou elektrický náboj, a proto přímo vyřážejí nebo vytrhávají Coulombickými elektrickými silami elektrony z atomů. Řadí se zde záření  $\beta^+$ ,  $\beta^-$ ,  $\alpha$  a protonové záření.
- **Nepřímo ionizující** – „jeho kvanta nejsou elektricky nabitá; svou kinetickou energii předávají v látce nejprve nabitým částicím a ty teprve přímými účinky na atomy látku ionizují. Patří zde především rentgenové záření a záření  $\gamma$ . [1]

**Dělení na záření vlnové a korpuskulární:**

Je známo, že se záření může chovat buď jako vlnění nebo proud částic. Kritérium rozhodující o vlnové či korpuskulární podstatě je **klidová hmotnost  $m_0$ <sup>1</sup>** kvant záření.

- **Záření vlnové** -  $m_0 = 0$ ; jedná se o kvanta vlnění, která se pohybují rychlostí světla, pokud jsou zabrzděna, předají veškerou energii a zaniknou. Do této skupiny se řadí vlnění elektromagnetické, jehož kvanta jsou fotony.
- **Záření korpuskulární** -  $m_0 > 0$ ; v tomto případě se jedná o proud hmotných částic pohybujících se rychlostí menší než rychlost světla. Po zastavení pohybu je zachována jejich existence. Do této skupiny se řadí  $\beta^+$ ,  $\beta^-$ ,  $\alpha$  a protonové záření. [1]

### 2.1.2. Interakce ionizujícího záření při průchodu hmotou

„Interakce záření s látkou probíhá primárně na **atomární úrovni**, příp. při vyšších energiích na jaderné a částicové úrovni.“ [1] Na rozdíl od makroskopických těles a elektromagnetických záření delších vlnových délek, které interagují **kolektivně** (současně) s mnoha atomy a molekulami, vysokoenergetická kvanta o velmi krátké vlnové délce interaguje **individuálně** s jednotlivými atomy látky, příp. atomovými jádry a elementárními částicemi. [1]

---

<sup>1</sup> Klidová hmotnost je měřená v inerciální vztažné soustavě, v níž je částice v klidu. [1]



### **Mechanismy uplatňující se při průchodu záření hmotou:**

- **Bez interakce** – kvantum záření volně proletí mezi atomy látky, nejčastěji k tomuto případu dochází při průletu tvrdého záření látkou o nízké hustotě.
- **Silné interakce** – mohou vyvolávat rozptyl, jaderné reakce, při vysokých energiích interakce elementárních částic za vzniku částic nových (elektrony, pozitrony)
- **Slabé interakce** – při průchodu běžných druhů ionizujícího záření se uplatní jen zcela výjimečně.
- **Elektromagnetické interakce** – při interakci elektricky nabitých částic s elektromagnetickým polem, v němž se náboje nacházejí, vznikají kvanta záření, fotony, podle své energie většinou fotony rentgenového nebo gama-záření. [1]

Záření má v látce **omezený dolet/dosah**, protože všechny výše uvedené mechanismy způsobí, že při průletu záření látkou dochází ke ztrátám energie částic, k jejich brždění a nakonec i zastavení. „Podél dráhy svého průletu zanechávají kvanta záření ionizační stopu z volných záporných elektronů a kladných iontů. Část těchto iontů opět vzájemně rekombinuje, avšak jejich část může vyvolat nové chemické vazby a reakce v okolní látce.“ [1] V konkrétním případě u fotonového záření ( $X$ ,  $\gamma$ ) předají fotony veškerou svou energii částicím látky, většinou elektronům, a samy zaniknou při fotoefektu. [1]

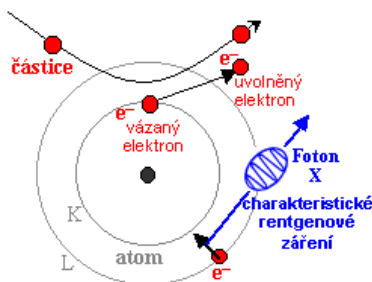
#### **2.1.2.1. Sekundární záření generované při interakcích záření s látkou**

Každý předmět ozářený primárním zářením se stává zdrojem sekundárního záření. Zde je uveden pouze výčet, konkrétně bude některým druhům věnována pozornost v následující kapitole: **brzdné záření**, **Comptonovsky rozptýlené  $\gamma$ -záření ( $X$  záření)**, rozptýlené neutrony, **fotoelektrony**, **charakteristické  $X$ -záření**, augerovy elektrony, elektronové a pozitronové záření, **anihilační  $\gamma$ -záření**, protony a neutrony, mezony, světelné záření. [1]

#### **2.1.3. Interakce nepřímo ionizujícího záření**

Interakce záření  $\gamma$  a  $X$  s látkou, která vede k ionizačním účinkům, může probíhat čtyřmi různými způsoby.

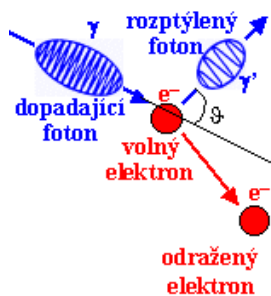
### 2.1.3.1. Fotoefekt



Obrázek 1 Fotoefekt, převzato z [2]

Jakmile dojde ke **srážce fotonu a elektronu**  $e^-$ , který je vázán v atomovém obalu, foton odevzdá veškerou energii a **zanikne**. Tato energie se spotřebuje na uvolnění elektronu z atomového obalu a na kinetickou energii vyraženého fotoelektronu. „Elektron, který získal tuto energii, se uvolní z vazby v atomu a **vyletí** s kinetickou energií, danou rozdílem energie záření a vazbové energie elektronu v atomu. Následně se bude fotoelektron pohybovat „cik-cak“ a odrážet se mezi atomy a **ionizovat** je až se nakonec zabrzdí. Na místo uprázdněné po elektronu okamžitě přeskočí z vyšší slupky v atomovém obalu elektron, přičemž energetický rozdíl vazbové energie na vyšší a nižší slupce se vyzáří ve formě kvanta (fotonu) elektromagnetického záření – **charakteristické rentgenové záření**.“ [1] Pravděpodobnost vzniku fotoefektu roste s rostoucí hustotou látky.

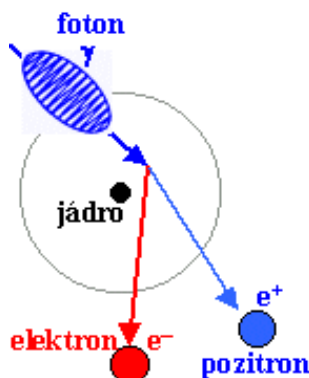
### 2.1.3.2. Comptonův rozptyl



Obrázek 2 Comptonův rozptyl, převzato z [2]

Comptonův rozptyl vzniká, pokud dojde ke srážce záření  $\gamma$  s volným elektronem, nebo elektronem slabě vázaným. Elektron přijme **část energie** fotonu, ten se od elektronu pružně odrazí a pokračuje ve svém pohybu ve **změněném směru s nižší energií**. „Energie rozptýleného fotonu je závislá na úhlu rozptylu  $\theta$  – čím větší úhel rozptylu, tím více energie foton ztratí (a tím více ji získá odražený elektron). Elektron se srážkou urychlí na příslušnou kinetickou energii a bude ionizovat podobně jako každé jiné záření beta.“ [1] „Pravděpodobnost Comptonova rozptylu roste s hmotnostní hustotou látky a protonovým číslem, klesá s rostoucí energií fotonů. Comptonův rozptyl se může několikrát opakovat, až foton buď látku opustí, nebo ztratí tolik energie, že zaniká fotoefektem na některé z elektronových hladin atomů.“ [1]

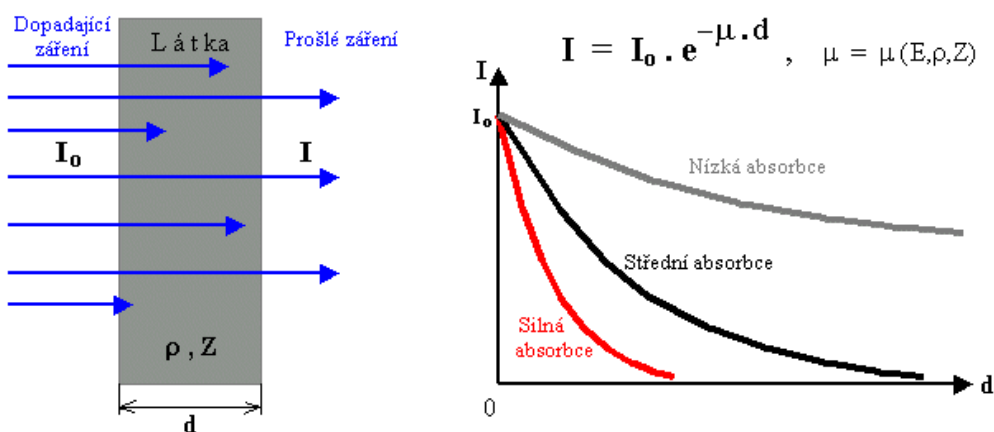
### 2.1.3.3. Tvorba elektron – pozitronových párů



Obrázek 3 Tvorba elektron – pozitronových párů, převzato z [2]

Průlet **vysokoenergetického** záření  $\gamma$  (1,022MeV) kolem atomového jádra se elektromagnetickou interakcí se silným Coulombovým polem může způsobit přeměnu fotonu  $\gamma$  na **dvojici** částic **elektron + pozitron**:  $\gamma \rightarrow e^- + e^+$ . Pouze elektron však zůstane v látce jako trvalá částice. Pozitron po zbrzdění podléhá anihilaci s některým z dalších elektronů a vznikají dva fotony záření  $\gamma$  o energii 511keV. Popsaný proces se nejčastěji vyskytuje v případě záření o vysoké energii a u látek s vysokým protonovým číslem. [1]

### 2.1.4. Absorpce záření v látkách



Obrázek 4 Základní zákonitosti absorpce ionizujícího záření v látce

Záření lze rozlišit z hlediska míry absorpce v látce na **málo pronikavé** a **pronikavé**. Zatím co u málo pronikavého záření se pohltí prakticky všechno záření, u pronikavého záření se část kvant pohltí a část projde. Jelikož je záření  $\gamma$  pronikavé, bude dále pozornost věnována jen jemu.

Na **Obrázek 4**, je znázorněna situace, kdy rovnoběžnému svazku záření  $\gamma$  o výchozí intenzitě  $I_0$ , postavíme do cesty vrstvu absorbující látky (s hustotou  $\rho$  a protonovým číslem  $Z$ ) o tloušťce  $d$ .

Část záření se absorbuje, intenzitu prošlého záření označíme **I**.“ [1] Množství prošlého záření závisí na tloušťce materiálu a to exponenciálně:

$$I = I_o \cdot e^{-\mu \cdot d} \quad (1)$$

Absorpční koeficient **μ** se nazývá **lineární součinitel zeslabení**. „Jeho hodnota závisí na hustotě a protonovém čísle absorpčního materiálu a výrazně také na energii záření ( $E_\gamma$ ). Lineární součinitel zeslabení je tím vyšší, čím vyšší je hustota a protonové číslo dané látky a tím nižší, čím je energie záření  $E_\gamma$  vyšší.“ [1] Např. při návrhu stínění je spíše než lineární součinitel zeslabení, důležitý parametr nazývaný polovrstva, udávající tloušťku vrstvy daného materiálu, která zeslabí intenzitu daného záření na polovinu a je popsána vztahem [1]:

$$d_{1/2} = \frac{\ln 2}{\mu} \quad (2)$$

#### 2.1.4.1. Stínění záření $\gamma$

V závislosti na aplikaci ionizujícího záření je většinou potřeba zabránit vniku záření do určitých míst nebo z určitých směrů (ochrana před ionizujícím zářením, zobrazovací metody – kolimace svazku záření). [1]

„Pro záření  $\gamma$  a X jsou nejvhodnějšími stínícími materiály látky s vysokou hustotou a protonovým číslem, tj. s vysokou elektronovou hustotou – především **olovo**, wolfram, uran. Používají se olověné kontejnery pro přepravu a skladování zářičů, zástěny z olověného plechu atd. Pro účinné odstínění záření  $\gamma$  o energii cca 100keV stačí vrstva tloušťky 2mm. Pokud je potřeba zachovat viditelnost, používá se **olovnaté sklo** s vysokým obsahem oxidu olova v tavenině.“ [1]

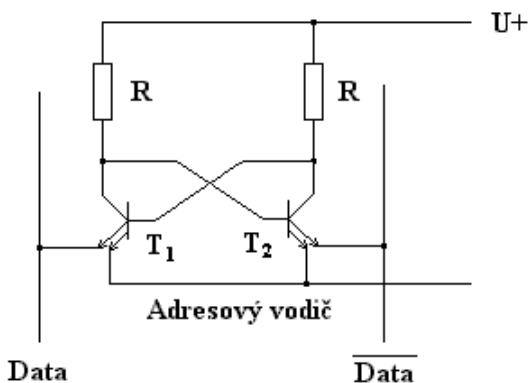
### 3. Výrobní technologie polovodičových pamětí

Výrobní technologie polovodičových součástek je jedním z faktorů ovlivňujících jejich odolnost vůči ionizujícímu záření, konkrétně mechanismy probíhající v materiálu při interakci s ionizujícím zařízením. Např. v integrovaných obvodech může vzniknout zkrat, v případě MOSFET se může nabíjet oxidová vrstva hradla a v paměťových součástkách existuje riziko narušení jednotlivých paměťových buněk. Pro výrobu polovodičových pamětí se používají dvě technologie – bipolární a unipolární.

#### 3.1. Bipolární technologie

Ke konstrukci paměťové buňky se používají bipolární tranzistory. Tyto paměti jsou sice rychlé, ale ve srovnání s unipolární technologií potřebují mnohem větší příkon. „Nedovolují dosáhnout vyššího stupně integrace, a proto mívají menší kapacitu než paměti unipolární. Používají se proto pouze pro speciální malé, ale velmi rychlé paměti, například vyrovnávací paměti“ [13]

Paměťové buňky jsou tvořeny obvody ECL<sup>2</sup> (Emitter – Coupled Logic) nebo TTL<sup>3</sup> (Tranzistor – tranzistorová logika). U bipolární paměti TTL je paměťová buňka obvykle bistabilní klopný obvod RS s víceemitorovými tranzistory. Tranzistory jsou vzájemně propojeny zpětným vazbami. Jeden z emitorů, obou tranzistorů je připojen na výběrový vodič a druhý emitor je u jednoho tranzistoru určen k zápisu a čtení logické hodnoty 1 a u druhého tranzistoru pro zápis a čtení logické hodnoty 0. [13]



**Obrázek 5** Realizace jedné buňky paměti SRAM v technologii TTL [32]

<sup>2</sup> Obvod ECL – „využívá tzv. proudové spínače, tj. zapojení tranzistorů s emitorovou vazbou. Tranzistory pracují výhradně v aktivní oblasti, a výsledkem je velmi malé zpoždění v logických členech. ECL obvody mají malý rozkmit signálů a tím nízkou šumovou imunitu.“ [14]

<sup>3</sup> Obvod TTL – základem je více vstupové hradlo NAND, jeho funkcí je zesílení a vlastní logická funkce. [14]



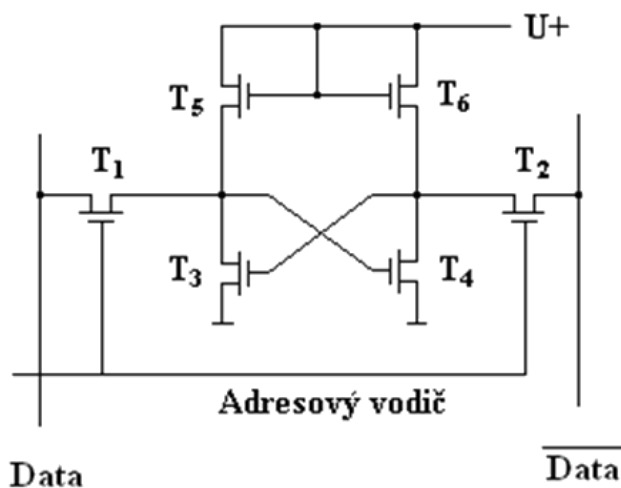
### 3.2. Unipolární technologie

Paměťové buňky jsou tvořeny unipolárními tranzistory MOS (Metal Oxide Semiconductor). Použití unipolární technologie nabízí možnost velmi vysokého stupně integrace. Unipolárních technologií je mnoho, odlišují se typem vodivosti kanálu tranzistoru, jeho morfologií, použitými materiály i výrobními postupy. Rozlišují se tři základní skupiny:

- P-MOS, technologie s  $p$  kanálem
- N-MOS, technologie s  $n$  kanálem
- CMOS, technologie s komplementárními tranzistory s  $p$  i  $n$  kanálem.

Základním prvkem CMOS obvodů je invertor tvořený komplementární dvojicí MOS tranzistorů s  $n$  a  $p$  kanálem. Dvojice invertorů  $T_3$ ,  $T_4$  a  $T_5$  a  $T_6$  tvoří bistabilní klopný obvod, do kterého lze zapsat informaci "0", nebo "1" po příslušném bitovém vodiči. Přitom musí být otevřený jeden ze čtecích, resp. zapisovacích tranzistorů  $T_1$  nebo  $T_2$  pomocí adresovacího vodiče pro výběr řádku, který je připojen k hradlům obou tranzistorů. [13] CMOS tranzistory je tvořena buňka statické paměti RWM.

„Základní paměťová buňka dynamických pamětí se vyrábí výhradně technologií MOS. Informace jednoho bitu je v ní uchována ve formě náboje v malé kapacitě, vytvořené vhodnou geometrií elektrod MOS tranzistoru.“ Kapacitor je nabíjen a vybíjen tranzistorem. „V klidovém stavu je adresový vodič na nulovém potenciálu a tranzistor  $T$  je otevřený. Kapacitor je buď nabitý (zápis "1") nebo vybitý (zápis "0").“ [13]



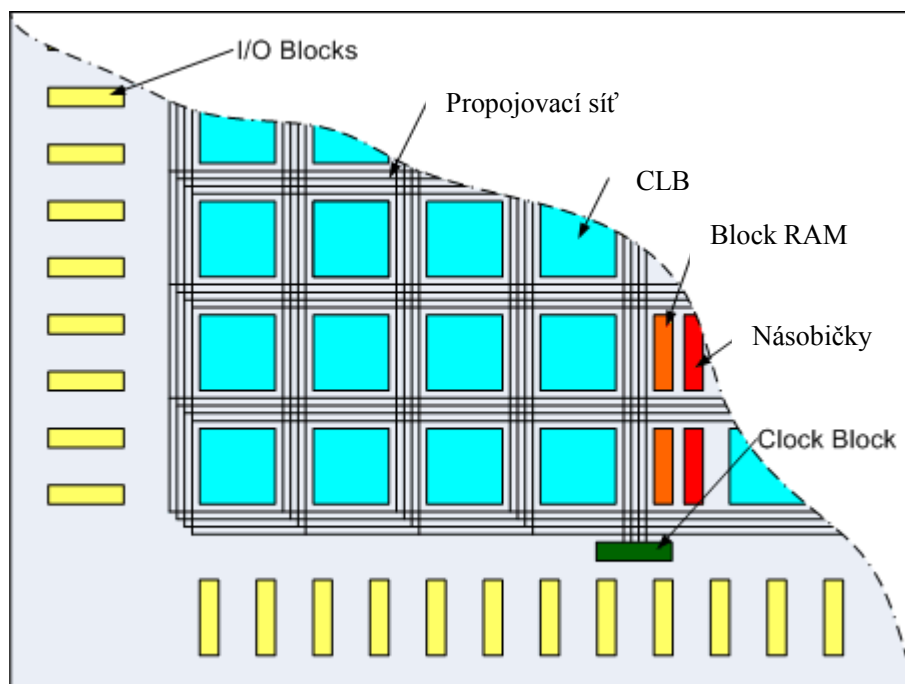
**Obrázek 6** Realizace jedné buňky SRAM v technologii MOS [32]

## 4. Technologie FPGA

„Zkratka FPGA vznikla z anglických slov Field Programmable Gate Array, což znamená programovatelná hradlová pole. FPGA tvoří podskupinu PLD (Programmable Logic Device) programovatelných logických součástek, do níž se dále řadí klasické PLD a komplexní PLD.“ [3] Obecně lze FPGA charakterizovat jako speciální integrované obvody s pravidelnou, zpravidla maticovou, strukturou logických buněk, schopných realizovat jednoduché logické operace. Propojením těchto buněk, pomocí konfigurovatelné propojovací sítě, lze získat rozsáhlé komplexní funkce, k jejichž realizaci by bylo zapotřebí mnoho různých obvodů. Propojovací síť tvoří mohutný systém vertikální a horizontálních vodičů. V místě křížení těchto vodičů jsou umístěny programovatelné propojky (spínače). [3], [5]

„V dnešní době je FPGA konkurentem zákaznickým integrovaným obvodům (ASIC – Application Specific Integrated Circuits). Hlavní předností FPGA vůči ASIC je programovatelnost, což znamená, že konfigurace obvodu FPGA je záležitostí uživatele (z čehož dále plyne opakovatelné využití), oproti ASIC, kde je obvod naprogramován již při výrobě. Koncept univerzálního předpřipraveného obvodu, s nímž bude možné realizovat mnoho různých úloh, je velmi populární, neboť je to neefektivnější způsob snižování výrobních nákladů.“ [3] FPGA nachází uplatnění v oblastech digitální techniky, kde je potřeba vysokého výpočetního výkonu, vysoké hustoty integrace, téměř neomezené flexibility a rekonfigurovatelnosti návrhu, podpory vysokorychlostní komunikace, možnosti rychlého číslicového zpracování signálů. [5]

### 4.1. Anatomie programovatelného obvodu



Obrázek 7 Základní architektura FPGA, převzato z [7]

**Základní architekturu FPGA tvoří:**

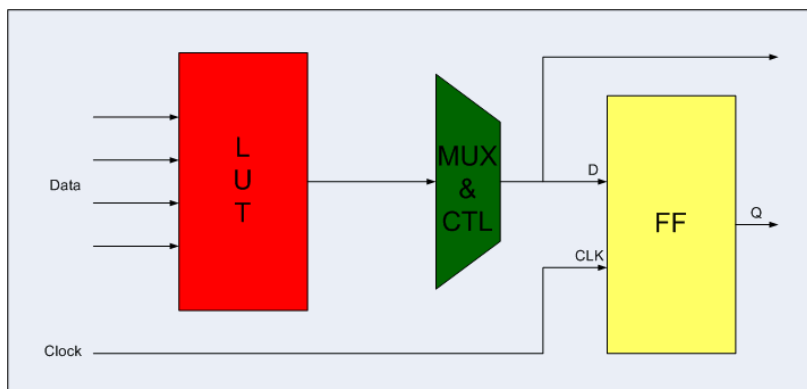
- Pravidelná matice konfigurovatelných logických bloků CLB (Configurable logic block)
- Vstupně/výstupní bloky IOB (Input/Output Blocks)
- Síť lokálních a globálních propojovacích vodičů
- Statická paměť konfigurace

**4.1.1. Konfigurovatelné logické bloky CLB**

„Každý blok CLB se skládá z několika (počet je závislý na architektuře) tzv. logických buněk (logic cell, slice, logic element).“ [5] Typická logická buňka obsahuje jednu nebo více **LUT (Look-Up Table)** klopné obvody, logiku přenosu a multiplexor, viz. **Obrázek 8**. LUT může realizovat jakoukoliv logickou funkci a zpravidla má čtyři vstupy a jeden výstup. Jedná se v podstatě o paměť, což znamená, že vnitřní zpoždění prvku LUT je konstantní, nezávislé na realizované logické funkci. Pokud je třeba většího počtu vstupů, nebo výstupů, lze zkombinovat více logických buněk v rámci CLB nebo více CLB bloků. „Kromě „generování logické funkce“ může LUT plnit u moderních obvodů i funkci paměti RAM, jejíž kapacitu lze zvyšovat spojováním více LUT. Taková paměť se nazývá distribuovaná (Distributed RAM), protože jednotlivé její části jsou rozprostřeny v architektuře.“ [5]. [7]

Dalším prvkem logické buňky je **klopný obvod typu D (Flip – Flop)**, zdrojem jeho hodinového signálu je krystalový oscilátor na čipu. Dřívější architektury FPGA umožňovaly taktovat klopné obvody hodinovým signálem, který byl vytvořen kombinační logikou, což způsobovalo asynchronní návrh, který sebou přinášel mnoho problémů. „Lze např. nastavit, na jakou hranu bude FF citlivý, zda bude mít vstup pro nulování nebo nastavení atd.“ [5]. [7]

**Multiplexor (MUX)** slouží k propojení signálů uvnitř logické buňky. „Logická buňka obsahuje také logiku přenosu a odpovídající signálové propojení k sousedním logickým buňkám. Tyto prostředky slouží k vytváření čítačů a dalších obvodů aritmetiky.“ [5]



**Obrázek 8** Zjednodušené schéma logické buňky, převzato z [7]

### 4.1.2. Vstupně – výstupní bloky I/O Blocks

IOB tvoří rozhraní mezi vnějšími vývody součástky (Package Pads) a signály vnitřní logiky. Jednotlivé bloky mohou být uživatelem konfigurovány jako vstupní, výstupní nebo obousměrné s třístavovým řízením.

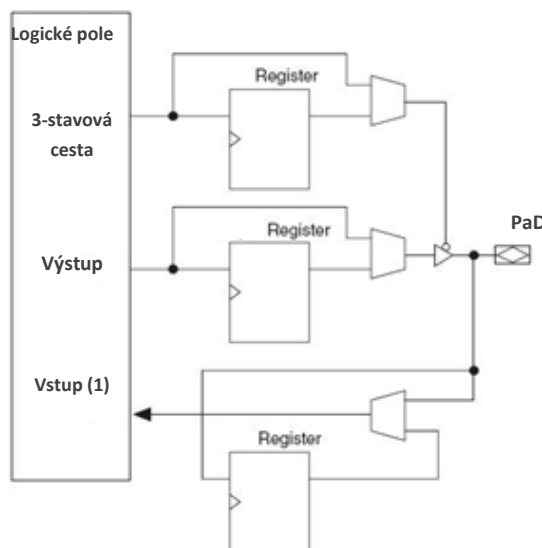
„**Vstupní signálová cesta** obsahuje především:

- Vstupní člen – input buffer/komparátor logické úrovně.
- Připojitelný Pull-Up resp. Pull-Down rezistor pro definování logické úrovně na nezapojeném vstupu.
- Klopný obvod typu D pro zachycení vstupního signálu

**Výstupní signálová cesta** obsahuje především:

- Klopný obvody typu D (registrový výstup)
- Programovatelný výstupní budič (Output Driver) s možností dvouhodnotového (SLOW/FAST) nastavení parametru Slow Rate (rychlost změny signálu z log “1” na log. “0” a naopak).

**Třístavová signálová cesta** obsahuje také klopný obvod typu D a slouží k řízení třístavového výstupního budiče.“ [5]



**Obrázek 9** Zjednodušený diagram vstupně/výstupního bloku, upraveno podle [7]

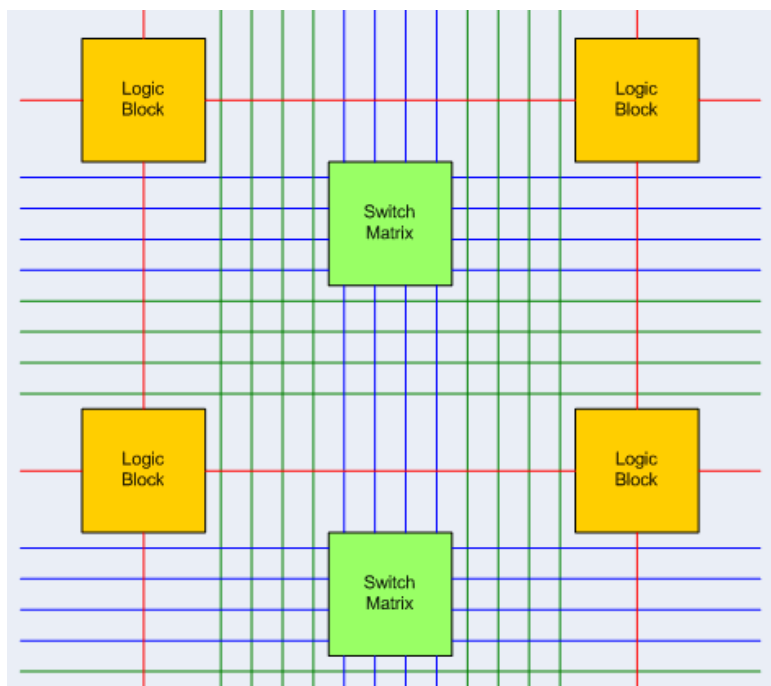
### 4.1.3. Síť propojovacích prostředků

Aby mohly být vedeny signály mezi CLB a IOB je třeba sítě propojovacích prostředků. Rozlišujeme tři způsoby propojení: lokální (local), programovatelná spínací matice (switch matrix) a dlouhé (long).

**Lokální propojení** slouží k propojení logického bloku s jeho nejbližším sousedním logickým blokem. Tímto způsobem lze získat komplexnější logické funkce, které by nebyl schopen realizovat jeden CLB.

**Programovatelná spínací matice** také propojuje logické bloky a umožňuje vést signály jak pod úhlem 90° tak pod 180° - v místě křížení dvou vodičů je 6 tranzistorů, jejichž konfigurace sepnutí nastavuje libovolný tvar signálové cesty - takto lze propojit CLB, které jsou relativně daleko od sebe. Nevýhodou je, že každý signál, který projde maticí je vystaven danému zpoždění. Jestliže ke zpoždění na signálu dojde mnohokrát, pak zpoždění ve vedení může být větší než skutečné zpoždění implementované logiky.

Třetím typem propojení je **dlouhé**. Slouží k propojení důležitých logických bloků, které jsou fyzicky daleko od sebe, tak aby nevznikalo velké zpoždění. Tyto cesty mohou být používány také jako sběrnice. [5], [7]



**Obrázek 10** Síť propojovacích prostředků, červenou barvou je označeno propojení lokální, modrou programovatelná propojovací matice a zelenou propojení dlouhé, upraveno podle [7]



## 5. Dosavadní poznatky ve vztahu ionizujícího záření a polovodičů

### 5.1. Účinky záření na základní polovodičové součástky

S ohledem na cíl práce, tedy zkoumání vlivu ionizujícího záření na polovodičové paměti, je třeba nejprve popsat teoretické předpoklady interakce základní struktury polovodičů s ionizujícím zářením, jak je uvedeno v odborné literatuře.

Vznik změn a poruch v charakteristikách ozařované paměti je závislý na typu radiace, rychlosti ukládání energie v polovodičové součástce, typu materiálu, ze kterého je polovodič vyroben nebo ze kterého je vyrobenou pouzdro polovodiče a konkrétní aplikaci. Radiační účinky lze rozdělit na dvě skupiny podle času, který potřebuje ozařované zařízení k obnově své funkce: **permanентní a přechodné**. [17]

Mezi **přechodné** účinky se řadí chybné signály (šum) či chybné logické stavy. Tyto změny nazývané lehké chyby (*soft errors*) nejsou destruktivní a obvykle zmizí po určitém krátkém časovém intervalu nebo mohou být odstraněny jednoduchým opatřením. **Permanентní** účinky neboli těžké chyby (*hard errors*) způsobují takové změny v polovodičové součástce, které nemohou být jednoduše odstraněny. Někdy jsou tyto změny zvrátelné, ale až po delším časovém intervalu a vyžadují odstavení součástky. [17]

#### 5.1.1. Základní mechanismy poškození polovodiče

Vznik nežádoucího mechanismu vychází z vlastností zodpovídající za mnoho důležitých vlastností mikroelektroniky – citlivost elektrických charakteristik polovodičů na **koncentrace příměsí nebo nepravidelnost** v materiálu, ze kterého je polovodič vyroben. Což sebou také nese odpovídající citlivost na koncentraci **defektů v krystalové struktuře**, které vznikají v důsledku interakcí s ionizujícími částicemi. Malé rozměry každého individuálního tranzistoru nebo paměťové buňky v integrovaném obvodu představují možnost narušení jedinou ionizující částicí. Integrované výrobní techniky produkují taková zařízení, jejichž **funkčnost závisí na každém elementu** z miliónu např., proto i velmi malá úroveň radiace ovlivňující jen malou část elementů, může způsobit jak nesprávné chování obvodu, tak selhání celého systému. [17]

- **Dislokace v krystalové mřížce**, v důsledku interakce s neutrony, protony, alfa částicemi a vysokoenergetickými gama fotony. Změny uspořádání atomů v krystalové mřížce způsobují trvalé poškození, zvyšují počet rekombinačních center, vyčerpávají minoritní nosiče (díry) a zhoršují vlastnosti polovodičových přechodů. Tento problém se projeví především u bipolárních tranzistorů, které závisejí na minoritních nosičích v oblasti báze. Výsledkem většího počtu rekombinací je ztráta schopnosti tranzistoru zesilovat. [19]

- **Ionizační efekty**, zapříčiněné nabitými částicemi. Účinky jsou často přechodné, způsobují většinou nepředvídatelné výstupy digitálních obvodů a lehké chyby, ale jakmile jsou spouštěčem jiného mechanismu (např. mikropřerazu<sup>4</sup>), mohou vést i k destrukci zařízení. Pokud je dávka dost vysoká, postupná akumulace děr v oxidové vrstvě MOSFET transistorů vede k zhoršení výkonu až k selhání zařízení. [19]

### 5.1.2. Následky interakce s ionizujícím zářením

Existují účinky, které závisí na množství radiační energie vyzářené na jednotku hmotnosti, např. celková přijatá dávka, a účinky jejichž dopad závisí na množství záření absorbované za jednotku času. Posledně zmíněný efekt je částečně spojen s pulzními zdroji energie, jako jsou například lineární urychlovače. Paradoxně při značném množství záření absorbovaného za jednotku času avšak s velmi úzkým pulzem nedojde k takové akumulaci, aby vzniklo poškození, a zároveň významná dávka může být v průběhu času akumulována i při bezvýznamném množství záření absorbované za jednotku času. [18]

- **Účinky neutronů** - neutronové zářiče uvolňují atomy do intersticiálních prostorů v krystalové struktuře a zároveň způsobují ionizaci<sup>5</sup>. [18]
- **Celková přijatá dávka** – kumulativní poškození krystalové mřížky. Vzniklé páry elektron – díra v izolační vrstvě báze indukují při rekombinaci fotoproud<sup>6</sup>, díry zachycené v defektech mřížky ovlivňují prahové napětí tranzistoru. Fotoproud vznikající v diodách, tranzistorech a integrovaných obvodech při dostatečně vysokých dávkách záření, představuje pro mnoho diod nebo tranzistorových struktur riziko zkreslení normálních signálů a tím přechodné narušení stavu elektronického zařízení. Většina zařízení následně obnoví normální funkci, ale při narušení stavů paměťových zařízení, jejichž výstupy závisí na stavech v předchozím čase, může dojít k trvalé ztrátě dat. V dalším případě je možné, že v malých částech nebo vodičích pulzy fotoproudu indukují tolik tepla, že dojde k jejich roztavení nebo vaporizaci. Tento jev se nazývá vyhoření (*burn-out*) a znamená kompletní nefunkčnost celého zařízení. [18] [19]
- **Přechodné účinky** – interakce jediného ionu s čipem může změnit stav paměti nebo bitových registrů. Nevzniká sice trvalé poškození, ale problémy systému, který se z chyby nedokáže obnovit. Ve velmi citlivých systémech je jeden ion schopen narušit až několik blízkých paměťových buněk. [19]

---

<sup>4</sup> Mikropřeraz – jedná se o fenomén, který vzniká mezi napájecími přívody v jádře CMOS obvodu. V podstatě mezi napájecími svorkami vznikne parazitický tyristor, který se vytvoří přímo na přechodech CMOS substrátu a vyskakuje napájení, takže CMOS obvod nedostane téměř žádný proud. [24]

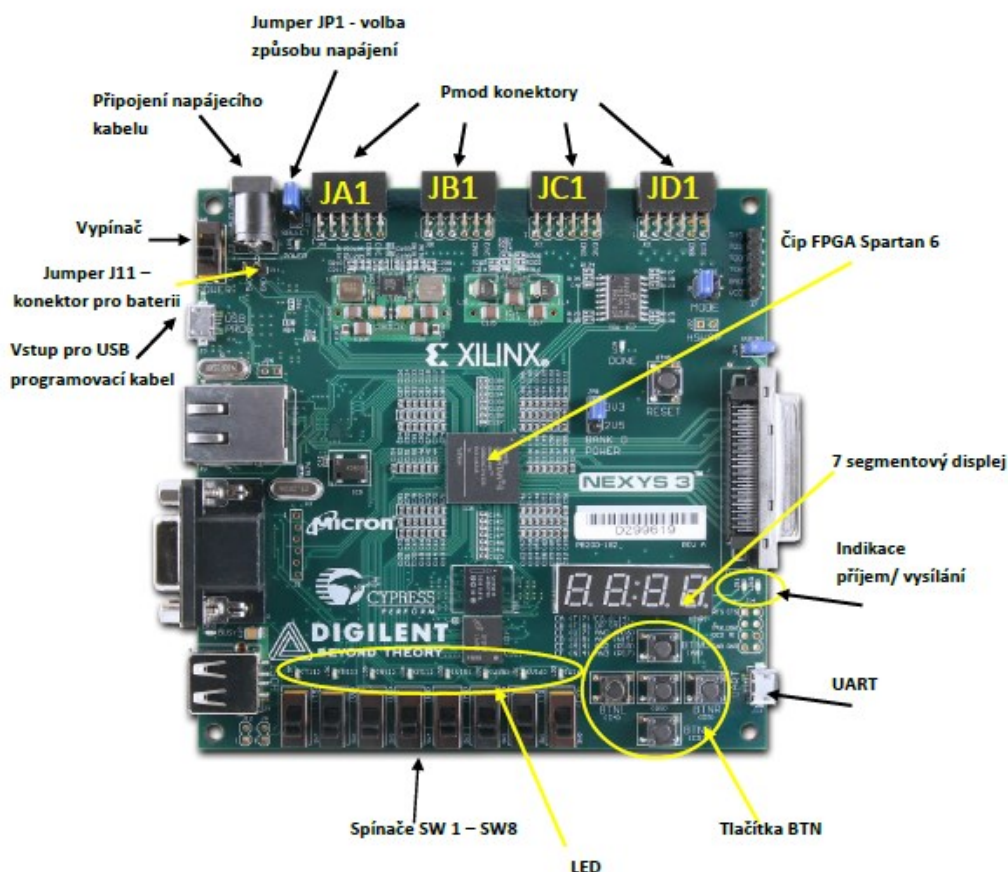
<sup>5</sup> Ionizace – „proces, při kterém se z elektricky neutrálního atomu nebo molekuly stává ion.“ [16]

<sup>6</sup> Fotoproud – elektrický proud tekoucí fotosenzitivním zařízením (fotodiody), nebo následek expozice zářivému toku. [20]

## 6. Návrh koncepce přípravku

K provedení analýzy vlivu ionizujícího záření na polovodičové paměti se jeví vhodné použití různých typů pamětí, ze skupin **EPROM**, **EEPROM**, **statické RAM**. Při výběru součástky hrála roli **velikost paměti**, **dostupnost na trhu** a **typ rozhraní** – paralelní nebo sériové. Oba typy rozhraní mají své výhody i nevýhody. Jednoduchost komunikace s paralelním rozhraním je nespornou výhodou, oproti tomu velký počet vstupů a s tím spojená šířka propojovacího kabelu může představovat komplikace. Sériové rozhraní má menší počet vstupů, ale průběh komunikace je složitější. Po zvážení všech aspektů bylo zvoleno jako kompromisní řešení použití komponent, jejichž technické specifikaci je věnována pozornost v následující podkapitole.

K implementaci obvodové logiky přípravku byla díky svým možnostem a dostupným periferiím vybrána deska FPGA NEXYS3 firmy Xilinx (**Obrázek 11** Vývojová deska NEXYS 3, upraveno podle [12]). Podrobná specifikace desky FPGA je uvedena v **kapitole 7**.



Obrázek 11 Vývojová deska NEXYS 3, upraveno podle [12]

## 6.1 Sledované paměťové obvody

### 6.1.1. EPROM

Paměťová součástka EPROM patří do skupiny pevných (permanentních) pamětí, nejčastěji označovaných ROM (Read Only Memory). Velkou výhodou ROM pamětí je, že při vypnutí napájecího napětí se obsah paměti neztrácí. EPROM (Erasable PROM) může být naprogramována uživatelem a pomocí ultrafialového záření také smazána. Jako paměťové buňky se používají **tranzistory MOSFET** opatřené plovoucím hradlem. [13]

#### EPROM 27C64A

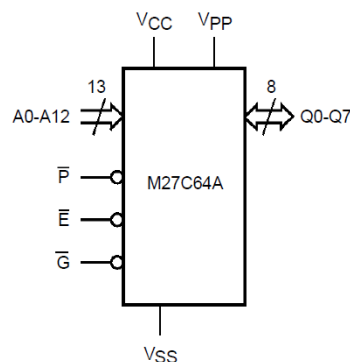
Součástka 27C64A je vysokorychlostní, ultrafialovým zářením mazatelná a elektricky programovatelná paměť s paralelním přístupem, tedy komunikace s pamětí je jednodušší než u pamětí se sériovým přístupem. Součástka má 28 pinové pouzdro (Ceramic Frit-Seal Dual-in-Line) s průhledným víčkem, které umožňuje vystavit čip UV záření.

Velikost	Organizace	Přístupová doba	Doba programování	Napájecí napětí	Pouzdro
65,536b	8192 slov po 8b	150ns	<1 minuta	-2 až 7V	DIP <sup>7</sup>

**Tabulka 1** Vlastnosti EPROM 27C64A, upraveno podle [10]

$A0-A12$	Adresové vstupy
$Q0-Q7$	Datové vstupy/výstupy
$\overline{E}$	Povolení vstupu
$\overline{G}$	Povolení výstupu
$\overline{P}$	Programování
$V_{PP}$	Programovací napětí
$V_{CC}$	Napájecí napětí
$V_{SS}$	Zem

**Tabulka 2** Názvy signálů



**Obrázek 12** Diagram logiky, převzato z [10]

<sup>7</sup> DIP (Dual-In-Line-Package) – čip je uložen v tvrdém plastovém nebo keramickém obalu, piny jsou umístěny podél delších stran pouzdra. Vývody procházejí skrz díry na DPS. [30]

**Proces mazání paměti:**

Mazání 27C64A začíná při vystavení buněk paměti světlu s vlnovou délkou kratší než 400nm. Tato hodnota odpovídá vlnové délce UV záření, sluneční světlo má vlnovou délku 300 – 400nm, rentgenové záření 0,01 – 10nm. Doba mazání je silně závislá na dávce záření (intenzita světla x doba expozice). Minimální hodnota dávky pro mazání je  $15 \frac{W}{cm^2}$  za 1s. Při

této dávce s použitím UV lampy s výkonem  $12000 \frac{\mu W}{cm^2}$  a vzdáleností 2,5cm od zdroje světla je doba mazání 15 – 20 minut. [10]

Jelikož se pro **naprogramování** paměti používá speciální programátor, tedy programovací logika nebude implementována do FPGA, nebudou programovací charakteristiky podrobně rozebrány.

**6.1.2. SRAM**

SRAM patří do skupiny pamětí RWM (Read Write Memory). Paměťová buňka je u statických RWM polovodičových pamětí tvořena **bistabilním klopným obvodem**. Složitost klopného obvodu závisí na použité technologii (bipolární, unipolární). Charakteristickou vlastností SRAM paměti je pouze dočasné uložení informace v paměťové buňce vázané na připojení napájecího napětí. Jednotlivé paměťové buňky jsou na čipu sestaveny do tvaru čtvercové matice, protože je tak nejlépe využít prostor uvnitř čipu. Aby mohla být vybrána libovolná z N paměťových buněk, je přivedená adresa dekodována pomocí řádkového a sloupcového dekodéru. Datový výstup paměťových součástek je konstruován jako třístavové hradlo, což umožňuje přenos dat v obou směrech po jedné sběrnici. Výstupy paměti se mohou nacházet v následujících třech stavech: logická '0', logická '1', stav vysoké impedance. [13]

**SRAM 6114**

Podobně jako v předchozím případě se jedná o paměť s paralelním přístupem. Předností této paměti je její velikost (1K x 4b), tedy i v případě vymazání celého obsahu paměti v průběhu analýzy, ukládání výsledků nebude tak náročné na velikost paměti uvnitř FPGA.

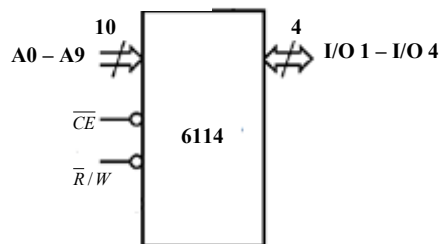
Velikost	Organizace	Přístupová doba	Napájecí napětí	Pouzdro
4096b	1024 slov po 4b	90ns	5V	DIP

**Tabulka 3** Vlastnosti SRAM 6114, upraveno podle [9]



$A0 - A9$	Adresové vstupy
$I/O1 - I/O4$	Datové výstupy
$\overline{CE}$	Povolení vstupu
$\overline{R}/W$	Čtení/zápis

**Tabulka 4** Názvy signálů



**Obrázek 13** Diagram logiky

### 6.1.3. EEPROM

EEPROM je elektricky mazatelná paměťová součástka a patří mezi pevné paměti. Před zápisem do paměti je smazán její aktuální obsah. Čtení z paměti probíhá mnohem rychleji než zápis. U těchto typů paměti je zdroj napětí pro generování programových napěťových úrovní i časovač určující délku programovacích impulzů umístěn na čipu paměti. Při programování slabiky nebo slova se pouze vloží adresa a požadovaná data, ty se uloží ve vnitřních záchytných registrech a proces programování probíhá samočinně na čipu. Výhodou EEPROM oproti EPROM je možnost programování paměti přímo v systému, bez nutnosti vyjmout paměť a použít speciální programátor. [13]

#### SPI Bus Serial EEPROM 25LC640

Hlavním důvodem volby 25LC640 k analýze byla její dostupnost na domácím trhu (jediná dostupná EEPROM s SPI rozhraním). Na rozdíl od předchozích dvou paralelních pamětí se nyní jedná o paměť se sériovým přístupem. Přístup k paměti je zprostředkován SPI kompatibilní sériovou sběrnici. Sériová paměť potřebuje ke své funkci hodinový signál, který slouží k synchronizaci komunikace master (FPGA) – slave (25LC640).

Velikost	Organizace	Doba zápisu	Napájecí napětí	Frekvence hodin	Pouzdro
64Kb	8192 slov po 8b	max. 5ms	2,5 – 5,5V	max. 2MHz	SOIC <sup>8</sup>

**Tabulka 5** Vlastnosti 25LC640, upraveno podle [11]

<sup>8</sup> SOIC (Small-Outline-IC) – pouzdro je menší a méně široké než ekvivalentní DIP pouzdro, na rozdíl od DIP je SOIC pouzdro určeno pro povrchovou montáž. [31]

<i>SCK</i>	Hodinový signál
<i>SI</i>	Sériový datový vstup
<i>SO</i>	Sériový datový výstup
$\overline{CS}$	Povolení vstupu
$\overline{WP}$	Ochrana zápisu
$\overline{HOLD}$	Pozastavení komunikace
<i>VCC</i>	Napájecí napětí
<i>VSS</i>	Zem

**Tabulka 6** Názvy signálů

### Princip funkce:

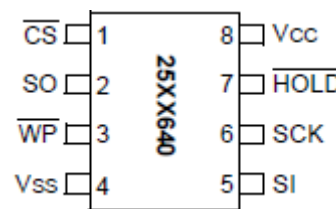
Paměť obsahuje 8 bitový registr instrukcí, viz **Tabulka 7**. Pokud je k zařízení povolen přístup a pin  $\overline{HOLD}$  je ve vysoké napěťové úrovni, lze načít s paměti komunikovat přes SI pin. Instrukce, adresy nebo data na pinu SI jsou vzorkována s náběžnou hranou hodin, zatím co data na pinu SO jsou aktualizována po sestupné hraně hodin. Přenos všech dat a instrukcí se řídí následující logikou: nejdříve je přenášen MSB a nakonec LSB. [11]

Název instrukce	Formát instrukce	Popis
<b>ČTENÍ</b>	0000 0011	čtení dat z paměti, začíná na zvolené adrese
<b>ZÁPIS</b>	0000 0010	zápis dat do paměti, začíná na zvolené adrese
<b>RDSR</b>	0000 0101	čtení stavového registru
<b>WREN</b>	0000 0110	Povolení operace zápisu

**Tabulka 7** Instrukční sady registru instrukcí, upraveno podle [11]

### Proces zápisu:

Proces zápisu začíná povolením přístupu k zařízení – nastavením  $\overline{CS}$  do log. „0“, následuje odeslání všech 8 bitů instrukce WREN a opětovné nastavení  $\overline{CS}$  do log. „1“, takto jsou nastaveny WE klopné obvody. Poté je možno opět povolit přístup k zařízení, odeslat instrukci pro zápis, následovanou adresou a nakonec daty. Poté co je odeslán LSB, musí být zakázán přístup k zařízení. Čtení z paměti během probíhající zápisu není možné. Poté co je zápis dokončen, jsou WE klopné obvody resetovány. [11]



**Obrázek 14** SOIC pouzdro – rozmístění pinů, převzato z [11]

**Čtení stavového registru:**

Čtení ze stavového registru se může provést kdykoli i během probíhající operace zápisu. Data ze stavového registru jsou formátována následovně:

7	6	5	4	3	2	1	0
WPEN	X	X	X	BP1	BP0	WEL	WIP

Pro účely této diplomové práce je důležitý pouze bit **WIP** (Write-In-Process), který indikuje, zda je paměť aktuálně zaneprázdněna probíhajícím zápisem, pak má hodnotu log. „1“.

**Proces čtení:**

Nejprve je nutné povolit přístup k zařízení. Poté je odeslána 8 bitová čtecí instrukce následována 16 bitovou adresou, jejíž 3 MSB jsou „don't care“ bity. Pokud je čtecí instrukce a adresa správně, data uložená na zvolené adrese jsou posunuta na SO pin. Data uložená na následující adrese mohou být čtena sekvenčně, pokud je dále zařízení poskytován hodinový signál. Interní čítač adres je inkrementován automaticky, poté co je odeslán každý datový byte. Když se dosáhne maximální hodnoty, dojde k přetečení a začíná se znovu od první adresy. [11]

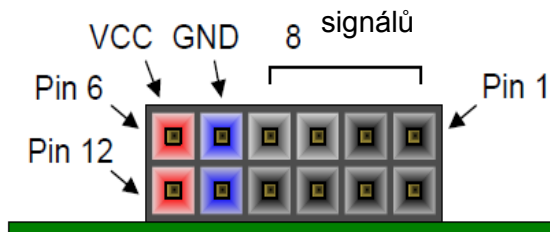
## 6.2 Návrh přípravku

Cílem projektu je realizovat přípravek, který je po propojení schopen komunikovat s deskou FPGA, ve které je implementována jeho obvodová logika a neustále, periodicky číst obsah sledovaných pamětí. V případě, že dojde ke změně v obsahu paměti, musí být tato změna zaznamenána s přesným časem. Frekvence výskytu změn může být vysoká, proto i samotná logika musí pracovat velmi rychle.

Přípravek bude s deskou NEXYS3 propojen pomocí dvouřadého **konektoru K s 32 piny**. Ke konektoru je připojen 32 žilový **plochý kabel** délky přibližně 1,5m. Kabel zvolené délky je použit z důvodu zajištění dostatečné vzdálenosti mezi zdrojem ionizujícího záření a deskou FPGA, aby nedošlo k jejímu **případnému poškození**. Při použití vodičů takové délky ovšem existuje riziko změny původně obdélníkového průběhu signálu vlivem parazitních kapacit a indukčností. Na konci kabelu jsou připojeny čtyři **dvouřadé konektory K1 – K4** s různým počtem pinů, které lze přímo připojit ke **konektorům FPGA desky – Pmod konektory (Obrázek 155)**.

Jak bylo zmíněno v předchozí podkapitole, k analýze byly vybrány dvě paměti s paralelním rozhraním a **DIP pouzdem** (EPROM, SRAM) a jedna se sériovým rozhraním a **SOIC pouzdem** (EEPROM). Pro účely měření je vhodné, aby bylo možné paměti z přípravku podle potřeby vyjmout, ať už kvůli programování/čtení u EPROM nebo při poškození některé součástky. Z toho důvodu jsou na přípravu připájeny **patice**, do kterých se paměti s DIP pouzdem umístí. EEPROM je na přípravku připájena přímo.

### 6.2.1. Pmod konektory



**Obrázek 15** Pmod konektor, pohled zepředu, upraveno podle [11]

Deska NEXYS3 má k dispozici celkem 4 Pmod konektory, označeny JA, JB, JC, JD (viz **Obrázek 1111**). Každý 12 pinový konektor poskytuje dva 3,3V Vcc signály (pin 6 a pin 12), dva GND signály (pin 5 a pin 11) a 8 logických signálů. Vcc a GND piny mohou poskytovat proud o hodnotě až 1A. Datové signály Pmod konektorů netvoří páry.

**Signálovými piny** Pmod konektorů jsou vedeny **signály** z/do desky FPGA **pro**:

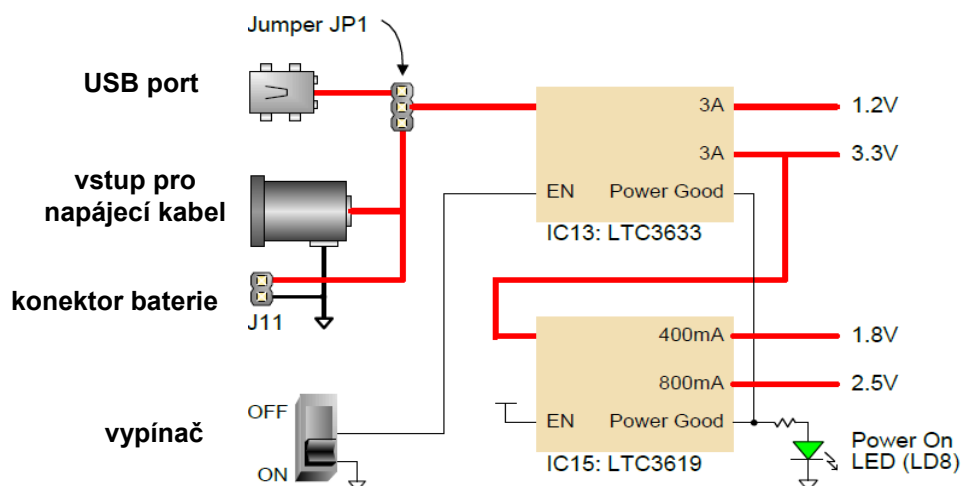
- EPROM: adresové vstupy, datové vstupy/výstupy.
- SRAM: adresové vstupy, datové vstupy/výstupy, povolení čtení a zápisu.
- EEPROM: CS pin, SI pin, SO pin, SCK pin.

**Vcc** signál konektoru JC 3,3V je použit pro **napájení** paměti **EEPROM**. Na JC konektoru se rovněž využívá **GND** pin jako **společný zemnicí signál** pro všechny paměti.

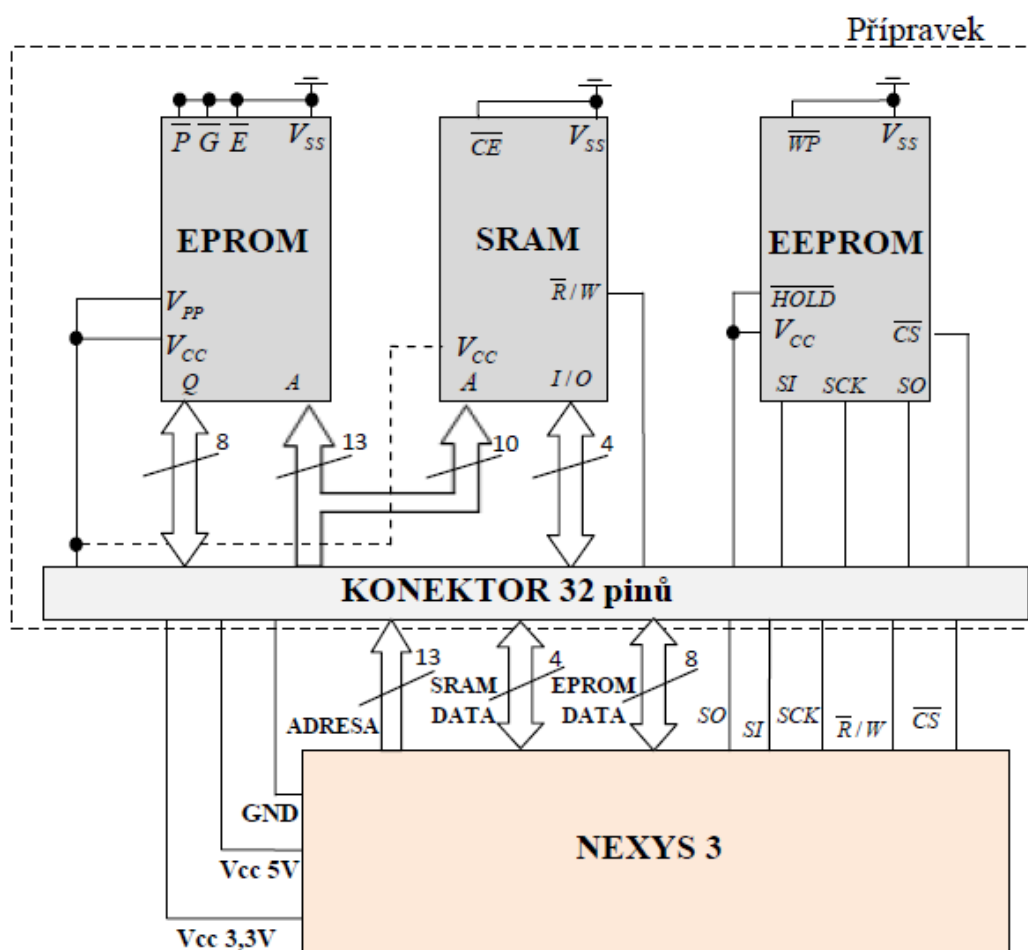
### 6.2.2. Napájení

Deska NEXYS3 může být napájena jak z USB vstupu pro programovací kabel, tak z externího zdroje – napájecí kabel připojený do elektrické sítě nebo z baterie. Způsob zapojení propojky JP1 určuje druh napájení (viz **Obrázek 16**). Maximální hodnota napájecího napětí je 5V DC. Propojka J11 slouží k připojení vodičů baterie. Díky obvodům napěťové regulace lze získat 3.3V, 2.5V, 1.8V a 1.2V zdroje z 5V napájecího napětí. [12]

Pro napájení pamětí SRAM a EPROM je z **5V** pinu konektoru pro baterii vedeno napětí přímo na jednu žílu plochého kabelu.



Obrázek 16 Napájení desky NEXYS 3, upraveno podle [11]



Obrázek 17 Blokové schéma přípravku, včetně propojení s deskou NEXYS3

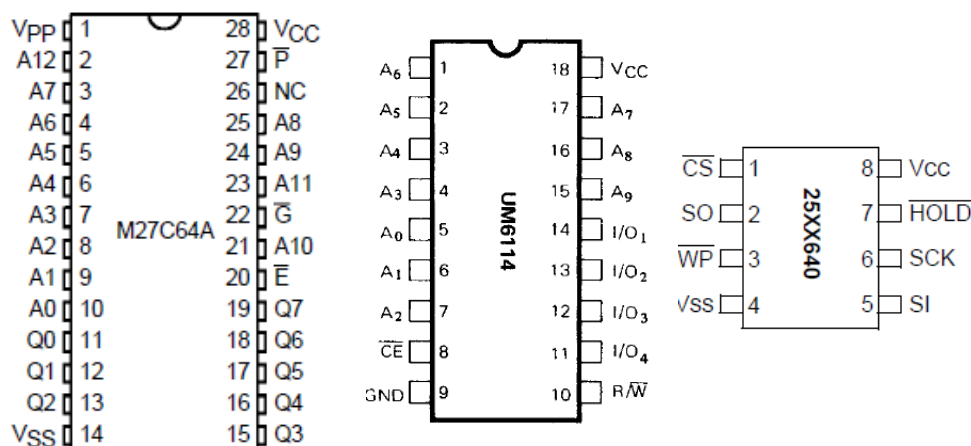
Konektor K přípravku	Kontakt	Vývod FPGA	Význam
1	K1(4)	JD1(4)	EEPROM: CS
2	JP1	JP1	EPROM: $V_{cc}$ , $\bar{P}$ , $V_{pp}$ ; SRAM: $V_{cc}$
3	K1 (10)	JD1 (3)	EPROM: Q3
4	K1 (3)	JD1 (10)	EPROM: Q4
5	K1 (9)	JD1 (2)	EPROM: Q5
6	K1 (2)	JD1 (9)	EPROM: Q6
7	K1 (8)	JD1 (1)	EPROM: Q7
8	K1 (1)	JD1 (8)	EPROM: A10, SRAM: A9
9	K1 (7)	JC1 (4)	EPROM: A11, SRAM: A8
10	K2 (4)	JD1 (7)	EPROM: A8, SRAM: A7
11	K2 $V_{cc}$	JC1 $V_{cc}$ (6)	EEPROM: $V_{cc}$
12	K2 GND	JC1 GND (5)	SRAM: $V_{ss}$ , CE; EEPROM: GND
13	K2 (10)	JC1 (3)	SRAM: R/W
14	K2 (3)	JC1 (10)	SRAM: I/O4
15	K2 (9)	JC1 (2)	SRAM: I/O3
16	K2 (2)	JC1 (9)	SRAM: I/O2
17	K2 (8)	JC1 (1)	SRAM: I/O1
18	K2 (1)	JC1 (8)	EEPROM: SI
19	K2 (7)	JB1 (4)	EEPROM: CLK
20	K3 (4)	JC1 (7)	EEPROM: SO
21	K3(10)	JB1 (3)	EPROM: A12, SRAM: A6
22	K3 (3)	JB1 (10)	EPROM: A7, SRAM: A5
23	K3 (9)	JB1 (2)	EPROM: A6, SRAM: A4
24	K3 (2)	JB1 (9)	EPROM: A5, SRAM: A3
25	K3 (8)	JB1 (1)	EPROM: A4, SRAM: A0
26	K3 (1)	JB1 (8)	EPROM: A3, SRAM: A1
27	K3 (7)	JA1 (4)	EPROM: A2, SRAM: A2
28	K4 (4)	JB1 (7)	EPROM: A1
29	K4(10)	JA1 (3)	EPROM: A0
30	K4 (3)	JA1 (10)	EPROM: Q0
31	K4 (9)	JA1 (2)	EPROM: Q1
32	K4 (2)	JA1 (2)	EPROM: Q2

**Tabulka 8** Propojení jednotlivých konektorů a význam jimi vedených signálů

## 6.3 Návrh a realizace DPS pro navržený přípravek

### 6.3.1. Návrh DPS

Deska plošných spojů byla navržena za použití software Eagle 6.3.0. Některé součástky byly **dostupné v knihovnách** (rezistory, kondenzátory, SOIC pouzdro paměti EEPROM), zbylé součástky byly **navrženy** pomocí editoru programu EAGLE 6.3.0 (pouzdro DIP paměti SRAM a EPROM, rezistorové sítě, dvouřadý konektor). Vzhledem k celkovému počtu vývodů pamětí – 54 a jejich rozdílnému uspořádání (**Obrázek 1818**) u jednotlivých součástek, bylo nutné návrh realizovat jako **oboustrannou** DPS. Na obou stranách DPS je rozlita zem, především z důvodu stínění.



**Obrázek 18** Rozmístění pinů pamětí, zleva EPROM, SRAM, EEPROM, převzato z [9] [10] [11]

### Způsob zapojení vývodů EPROM/SRAM

Vychází se z instrukcí uvedených v datových listech analyzovaných součástek.

- **Adresové vstupy (A<sub>0</sub> – A<sub>12</sub>/A<sub>0</sub> – A<sub>10</sub>)** - paměti EPROM a SRAM mají paralelní přístup a adresová sběrnice je pouze jednosměrná (z FPGA k paměti), je tedy možné, aby tyto paměti adresovou sběrnici **sdílely**. Adresové vývody jsou přímo propojeny s odpovídajícími piny konektoru K (viz **Tabulka 8** Propojení jednotlivých konektorů a význam jimi vedených signálů). U paměti EPROM slouží vývod **A<sub>9</sub>** pouze k vyčtení informací o výrobci a typu paměti, pro projekt nepodstatné, proto je pin trvale v log. „1“.
- **Datové vstupy/výstupy (Q<sub>0</sub> – Q<sub>7</sub>, I/O<sub>1</sub> – I/O<sub>4</sub>)** – jsou připojeny přes **odporový dělič** (viz níže) k odpovídajícím pinům konektoru K.
- **Napájení (V<sub>cc</sub>), zem (V<sub>ss</sub>/GND)** – napětí 5V i zemnicí signál jsou k pamětem přivedeny přes konektor K. Mezi napájení a zem, co nejbliž k napájecímu vývodu, je umístěn **blokovací kondenzátor** (viz níže).

- **Programovací napětí ( $V_{pp}$ ), programování ( $\overline{P}$ )** – EPROM. Protože je programování realizováno pomocí speciálního programátoru, tyto piny se během čtení nepoužívají, je zde připojeno napájecí napětí 5V.
- **Povolení výstupu ( $\overline{G}$ ) a povolení čipu ( $\overline{E}$ )** – EPROM. Během čtení platí pro oba vývody, že musí být na nízké napěťové úrovni, proto je k nim přes odporový dělič, (odpory 300 $\Omega$  a 3k $\Omega$ ) přivedeno napětí 0,5V.
- **Povolení čipu ( $\overline{CE}$ )** – SRAM. Během čtení i zápisu do paměti musí být na nízké napěťové úrovni, proto je vývod trvale uzemněn.
- **Čtení zápis ( $R/\overline{W}$ )** – SRAM. Připojen k odpovídajícímu pinu konektoru K.

#### Způsob zapojení vývodů EEPROM

- **Chip Select ( $\overline{CS}$ ), sériový výstup (SO), sériový vstup (SI), sériové hodiny (SCK), napájení ( $V_{cc}$ ) a zem ( $V_{ss}$ )** - jsou připojeny k odpovídajícím pinům konektoru K (viz Tabulka 8).
- **Pozastavení komunikace ( $\overline{HOLD}$ )** – komunikace je pozastavena pokud je v log. „0“, ale takové funkce není pro projekt třeba, proto je na pin trvale připojeno napájecí napětí 3,3V.
- **Ochrana zápisu ( $\overline{WP}$ )** – funkce také není pro projekt zapotřebí, datový list uvádí, že je nevyužitý pin třeba trvale uzemnit a pak všechny operace budou probíhat normálně.

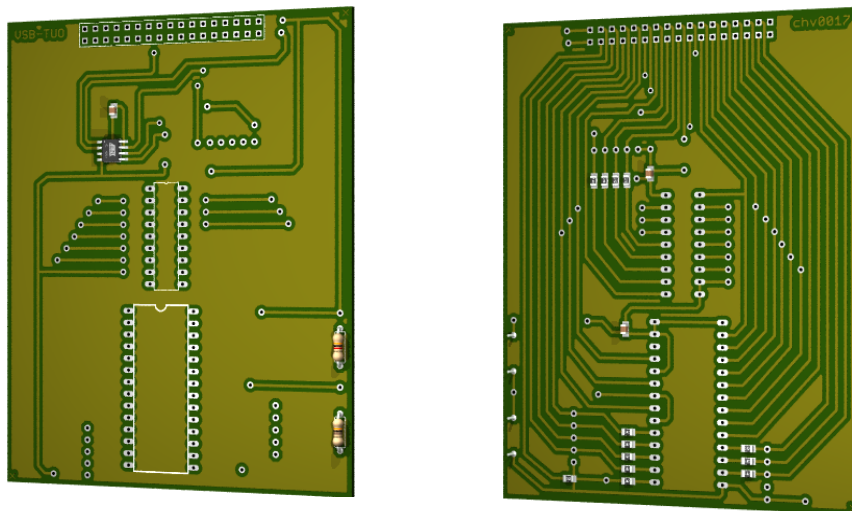
#### Blokovací kondenzátory

Blokovací kondenzátor potlačuje náhlé změny v napájení, realizuje tedy filtr napěťových špiček, který propouští pouze stejnosměrnou složku signálu. Digitální logické obvody, vyžadují pro správnou funkci stabilní napájení, jinak může dojít k neočekávanému chování takového obvodu. Milióny hradel, které tvoří logické obvody, neustále mění svůj výstupní stav na „zapnuto“ nebo „vypnuto“. Což znamená, že je v závislosti na frekvenci, během sekundy mnohokrát sepnuto a vypnuto několik tranzistorů. S každým sepnutím tranzistory produkují přechodné zatížení. Výsledkem je kolísání v odběru proudu zařízením a vznik šumu, který se šíří zpátky k napěťovému zdroji. V tom případě blokovací kondenzátor plní dva úkoly: chrání napájení proti elektrickému šumu, který vzniká uvnitř obvodu a obvod chrání proti šumu z ostatních zařízení, připojených k stejnému napěťovému zdroji. Neoptimálnější je umístit blokovací kondenzátor co nejbližší k napájecímu vývodu. [8] Konkrétně byly použity kondenzátory o jmenovité hodnotě **100nF**, doporučené výrobcem paměťových součástek.

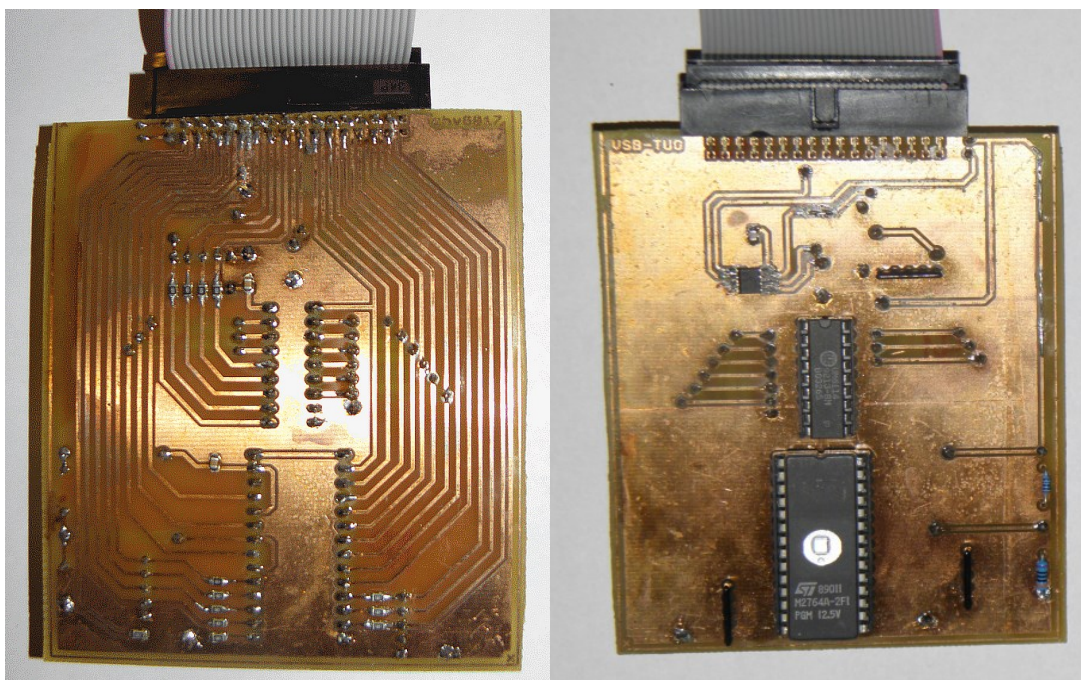


### Odporový dělič

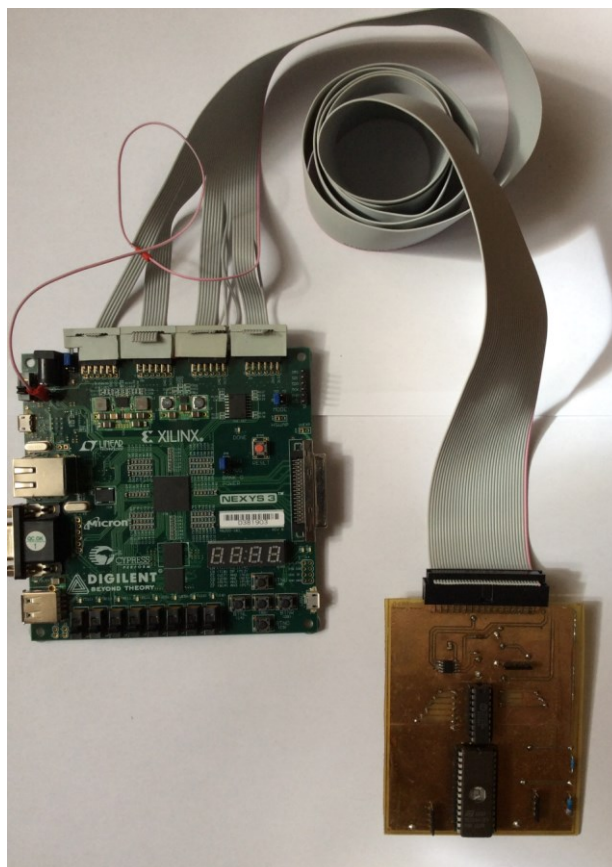
Deska FPGA poskytuje přípravku napájecí napětí 3,3V. Paměti SRAM a EPROM však potřebují k funkci napájecí napětí 5V. S tím také souvisí hodnota napětí na datovém vstupu/výstupu paměti. Deska FPGA s pamětí komunikuje přes obousměrnou linku a přivedením signálu s napětím vyšší než 3,3V by mohlo desku poškodit. Proto je v návrhu pomocí čtyř rezistorů o hodnotě **1k8** a rezistorové sítě o hodnotě **3k3** realizován odporový dělič, který dělí napětí 5V v poměru 3/5, do FPGA desky je veden signál o maximální hodnotě 3,3V.



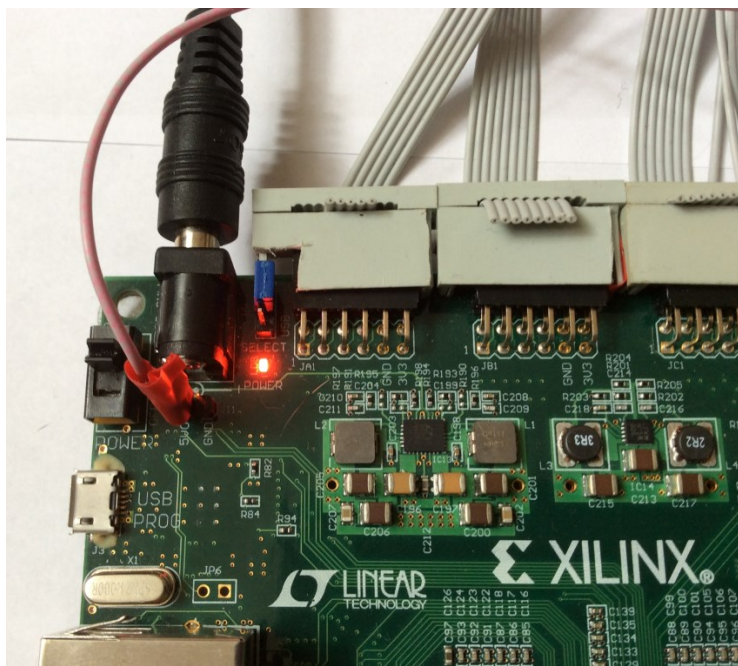
Obrázek 20 Realizace DPS, 3D model vygenerovaný programem Eagle



Obrázek 19 Realizace DPS



**Obrázek 21** Propojení přípravku s deskou NEXYS3



**Obrázek 22** Detail připojení napájecího napětí 5V

## 7. Vlastnosti FPGA řady Spartan6

FPGA řady Spartan6 představují moderní platformy, které v porovnání se staršími generacemi Spartan přinášejí o 65% menší spotřebu energie, vyšší výkon a více využitelných periferní. Kromě vyššího počtu logických buněk, až 576Kb velmi rychlé blokové paměti, nabízí Spartan6 také USB – UART port (3,2Mbps), 16Mb Cellular RAM, vylepšené vysokorychlostní konektory. Oproti generaci Spartan3, disponuje řada Spartan6 USB2 konektorem pro programování a přenos dat, není už tedy zapotřebí speciálního programovacího kabelu.

Počet logických buněk	Konfigurovatelné logické bloky (CLB)		CellularRAM (Mb)
	Vrstvy	Klopné obvody	
14,579	2,278	18,224	16
Bloky blokové paměti RAM		DSP 48A1 vrstvy	Max. uživatelských rozhraní I/O
18 Kb	Max (Kb)		
32	576	32	232

Tabulka 9 Přehled atributů Spartan6 XC6LX16

### 7.1 CLB, Vrstvy a LUT

Každý konfigurovatelný logický blok Spartan-6 FPGA obsahuje dvě vrstvy. Existují tři druhy CLB vrstev: SLICEM, SLICEL a SLICEX. Každá vrstva obsahuje čtyři LUT, osm klopných obvodů.

- **SLICEM** – každá ze čtyř LUT může být konfigurována jako 6-vstupá LUT s jedním výstupem, nebo jako duální 5-vstupá LUT s identickými 5 bitovými adresami a dvěma nezávislými výstupy. Tyto LUT mohou být také využity jako 64 bitová distribuovaná paměť RAM, 32 bitový posuvný registr, nebo dva 16 bitový posuvné registry. Klopné obvody uvnitř CLB jsou schopny registrovat každý výstup LUT. Pro aritmetické operace je zde vysokorychlostní přenosový řetězec.
- **SLICEL** – tyto vrstvy sdílí všechny vlastnosti s SLICEM, kromě paměťové funkce a možnosti konfigurace jako posuvný registr.
- **SLICEX** – vrstvy mají stejnou strukturu jako SLICEL kromě možnosti přenosu aritmetických operací.

## 7.2 Bloková paměť RAM

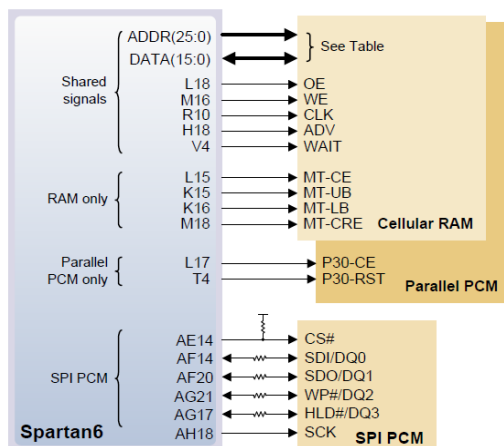
Bloková RAM je jednoúčelová paměť tvořena bloky statických pamětí uvnitř FPGA, kterou lze použít pro libovolný číslicový návrh. Blokovou paměť lze konfigurovat jako jednobránovou (jedna adresová a jedna datová sběrnice) nebo dvoubránovou (dvě adresové i datové sběrnice).

Spartan6 XC6LX16 obsahuje 32 dvoubránových blokových RAM, přičemž velikost jedné RAM je 18Kb (viz. **Tabulka 9**). Každý blok má dvě nezávislé brány, které sdílejí pouze uložená data. **Synchronní operace:** jakýkoli přístup k paměti, ať už se jedná o zápis nebo čtení je řízen hodinových signálem. [12]

## 7.3 CellularRAM

Jedná se o pseudo – statickou<sup>9</sup> DRAM paměť o velikosti 16Mb s 16 bitovou sběrnicí. CellularRAM může pracovat ve dvou režimech: **asynchronní a synchronní**. Když paměť pracuje v asynchronním režimu, pak cyklus zápisu a čtení trvá 70ns, přičemž probíhá automatický „refresh“ vnitřních DRAM polí. Pokud se nachází v synchronním režimu, pak pracuje na frekvenci 80MHz. Paměť se adresuje použitím horního a dolního bytu (MT-UB, MT-LB).

Pomocí dvou **řídících registrů** je definována funkce paměti: BCR (Bus Configuration Register) definuje jak CellularRAM komunikuje se sběrnicí systémové paměti a RCR (Refresh Configuration Register) řídí způsob „refresh“ DRAM polí. Oby tyto registry jsou automaticky nahrány do zařízení (s defaultním nastavením) při jeho zapnutí a mohou být aktualizovány kdykoli během funkce.



**Obrázek 23** Paměťové rozhraní, převzato z [12]

<sup>9</sup> Pseudo – statická DRAM – SRAM nepotřebuje neustále periodické obnovování, aby si udržela svá data, ale zato paměťová buňka je poměrně velká. Opačný případ platí pro DRAM, která sice potřebuje k udržení náboje na kondenzátoru periodické obnovování, ale zato je buňka malá a tedy hustota buněk na čipu může být větší. Pseudostatická paměť kombinuje výhody obou typů pamětí – paměťové buňky jsou dynamické a všechna potřebná obnovovací logika je integrována přímo na čipu, takže paměť funguje podobně jako SRAM. [23]

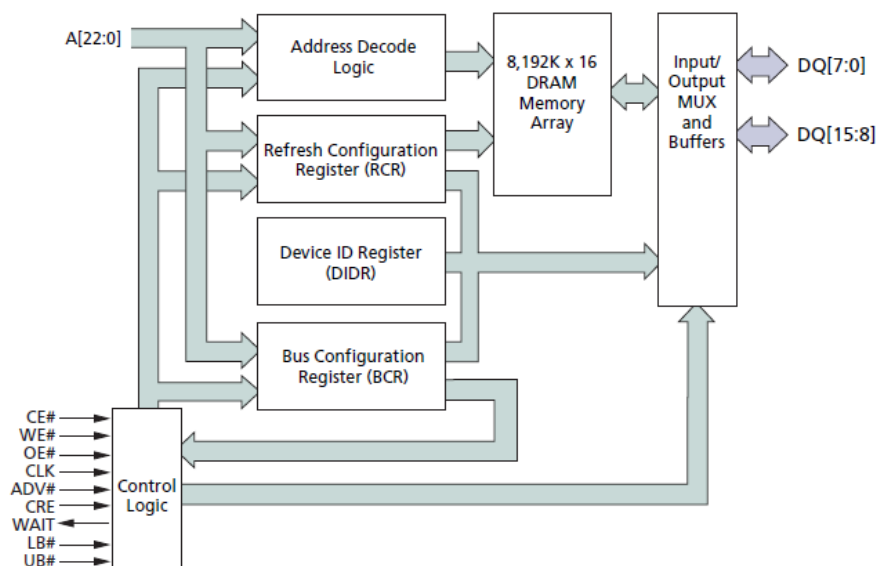
Symbol	Typ	Popis
A [22:0]	Vstup	Adresové vstupy: vstupy pro adresy během čtecí a zapisovací operace. Adresy jsou během operace interně inkrementovány.
ADV#	Vstup	Address valid: Indikuje platnou adresu na vstupu. Adresy jsou taktovány při náběžné hraně ADV# během asynchronního čtení a zápisu. ADV může být na nízké úrovni během asynchronních operací.
CRE	Vstup	Control register enable: Pokud je CRE na vysoké úrovni, operace zápisu je nahrána do RCR nebo BCR a operaci čtení je zpřístupněn RCR, BCR.
OE#	Vstup	Output enable: Zpřístupňuje výstupní buffery, když je na nízké úrovni. Pokud je na vysoké úrovni, k bufferům je přístup zamezen.
CE#	Vstup	Chip enable: Aktivuje zařízení, pokud je na nízké úrovni. Když je na vysoké úrovni, zařízení přechází do standby režimu.
WE#	Vstup	Write enable: Rozhoduje, zda je daný cyklus zapisovací. Pokud je WE# na nízké úrovni, cyklus je zapisovací jak do konfiguračního registru, tak do paměťového pole.
LB#	Vstup	Povolení spodního bytu. DQ [7:0]
UB#	Vstup	Povolení horního bytu. DQ [15:8]
DQ [15:0]	Vstup/ Výstup	Datové vstupy/výstupy.
WAIT	Výstup	Wait: Poskytuje zpětnou vazbu o platnosti dat během „dávkové“ čtecí a zapisovací instrukce. WAIT zabraňuje kolizi mezi refreshem a čtecí/zapisovací instrukcí. Když je CE# na vysoké úrovni WAIT je ve stavu vysoké impedance.

Tabulka 10 Význam vývodů CRAM

Asynchronní mód	Napájení	CLK	ADV#	CE#	OE#	WE#	CRE	LB# /UB#	DQ [15:0]
Čtení	Aktivní	L	L	L	L	H	L	L	Výst. data
Zápis	Aktivní	L	L	L	X	L	L	L	Vst. data
Nečinnost	Nečinnost	L	X	L	X	X	L	X	X

Tabulka 11 Pravdivostní tabulka CRAM





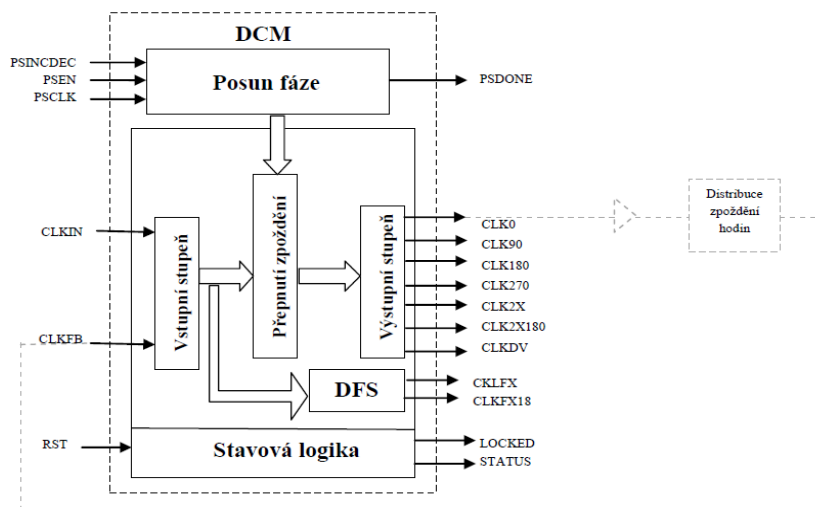
Obrázek 24 Funkční blokový diagram, zjednodušený popis funkce CRAM, převzato z [15]

## 7.4 Digital Clock Manager (DCM)

Blok DCM řídí, upravuje a distribuuje hodinový signál po celém čipu. Díky DCM lze eliminovat zkreslení hodinového signálu, čímž se zlepší výkon celého systému, dále umožňuje vytvořit novou hodinovou frekvenci vynásobením či dělením původní frekvence a filtrovat kolísání na hodinovém vstupu. [21]

DCM realizuje čtyři základní funkce: **závěs zpoždění** (Delay-Locked Loop DLL), **úprava kolísání** (Skew Adjustment), **frekvenční syntéza** (DFS) a **fázový posun**. Všechny funkce mohou pracovat jak samostatně, tak jako celek. Zjednodušený blokový diagram je uveden na Obrázek 25.

**Frekvenční syntéza** – lze vytvořit široké a flexibilní spektrum výstupních frekvencí, na základě poměru mezi multiplikátorem a dělitelem, definované uživatelem. Výstupní frekvence je pak výsledkem současného násobení a dělení vstupní frekvence. [21]



Obrázek 25 Blokový diagram DCM, upraveno podle [21]

## 8. Návrh obvodové logiky v jazyce VHDL

Vhodným nástrojem pro návrh obvodové logiky je jazyk **VHDL** (Very High Speed Integrated Circuits Hardware Description Language), který je jedním z nejpoužívanějších jazyků pro popis hardwarových struktur hradlových polí, umožňující návrh logických i sekvenčních struktur. Z výhod VHDL lze jmenovat **univerzálnost** – návrh je téměř nezávislý na cílové platformě, tzn. návrh je možno provádět pro hradlová pole různých výrobců, protože konečná implementace je závislá až na kompilaci VHDL kódu použitím vhodného kompilátoru. Jazyk je **dobře čitelný**, příkazy se dají okomentovat, návrhář si tak může během tvorby projektu dělat poznámky, a když se k návrhu po čase vrátí, lehce se v něm zorientuje. S jazykem VHDL pracuje **vývojové prostředí ISE Design Suite** firmy Xilinx, které je volně dostupné na internetových stránkách. Po vytvoření návrhu a jeho syntéze, umožňuje prostředí ISE dále provést **simulace** návrhu, nakonec i **implementaci** a **programování** cílového zařízení.

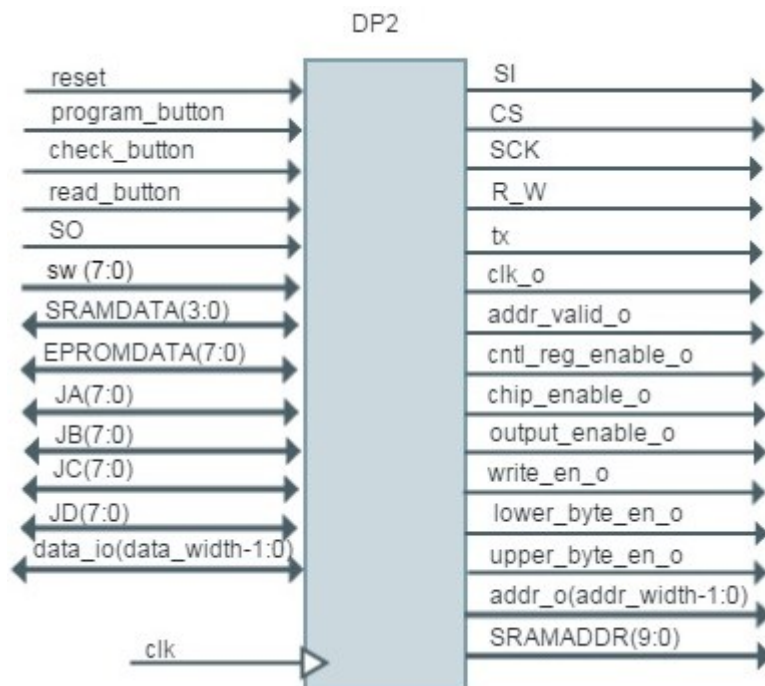
Při návrhu obvodové logiky FPGA přípravku se vycházelo z několika požadavků. Hlavní kritérium představuje obousměrná komunikace mezi přípravkem a deskou FPGA, jelikož se budou vybrané paměti pomocí obvodové logiky FPGA jak programovat, tak se z nich budou číst data. Byly vytyčeny následující **cíle**:

- **Programování** - SRAM a EEPROM stiskem jednoho tlačítka, zároveň uložení výchozího obsahu všech pamětí do vnitřní blokové paměti FPGA.
- **Kontrola** – stiskem tlačítka se spustí čas a začne probíhat porovnávání aktuálního obsahu pamětí během ozařování s výchozím obsahem pamětí. Pokud dojde ke změně dat jakékoli buňky paměti, záznam o události bude uložen do paměti výsledků na desce FPGA. Jakmile se zkontrolují všechny buňky pamětí, výchozí obsah pamětí uložený v blokové paměti se přepíše změněnými daty a další cykly kontroly budou probíhat na základě stejné logiky. Kontrola se ukončí opětovným stiskem tlačítka.
- **Záznam o chybě** – při detekci chyby bude vytvořen datový paket obsahující: typ paměti, čas s přesností na setiny sekundy, adresu smazané buňky a změněná data.
- **Odeslání výsledků** – po ukončení měření se veškerá data, uložená v paměti výsledků, odešlou pomocí sériové linky do osobního počítače.

Vzhledem k náročnosti vytyčených cílů se jako nejvhodnější řešení jeví vytvoření **hierarchického návrhu** (HD), který umožňuje rozčlenit návrh do bloků – modulů, aby se složitá konstrukce rozdělila na menší, lépe zvládnutelné části. Jednotlivé moduly se poté importují do modulu nejvyšší úrovně pomocí deklarace a instanciací komponenty. Modul je reprezentován entitou se vstupními, výstupními případně obousměrnými porty. Pro vstupní porty platí, že hodnoty signálu jsou určeny pouze ke čtení a do výstupních signálů lze pouze zapisovat. Obousměrné porty mohou být navíc ve stavu vysoké impedance. Signály deklarované v entitě, se nazývají **externí signály (porty)**, mohou být vedeny k ostatním modulům nebo z nich, případně přímo na piny desky a uvnitř architektury jsou k dispozici jako interní signály.

Mimo externích signálů se při návrhu využívá **interních signálů**, které jsou deklarovány v architektuře a jsou k dispozici pouze pro daný modul. V tomto návrhu každý modul představuje jeden **stavový automat**. Principy funkce modulů budou detailně rozebrány v následující podkapitole

## 8.1 DP2



Obrázek 26 Entita DP2

DP2 představuje „*top module*“ celého projektu, což znamená, že jsou zde deklarovány a instanciovány všechny ostatní komponenty návrhu. Než přistoupíme k popisu funkce DP2, je nutné nejprve definovat význam vstupních a výstupních portů.

### 7.1.1. Vstupní, obousměrné porty

- **Reset** – signál je řízen stiskem tlačítka BTNR na desce NEXYS3. Stiskem tlačítka reset, jsou všechny stavové automaty uvedeny do výchozího stavu (INIT), vynulovány všechny čítače a znemožněna komunikace se všemi paměťmi.
- **Program\_button (BTNS), check\_button (BTNU), read\_button (BTNU)** – stiskem jednoho z tlačítek se spouští požadovaný proces – programování, kontrola, odesílání dat do PC – pozornost jim bude věnována v samostatných kapitolách.
- **SO** – „*serial output*“ jeden z pinů Pmod konektoru (JC4), zprostředkovává komunikaci s výstupem EEPROM paměti.



- **Sw** – signály ze spínačů na desce NEXYS3. Kombinace logických signálů ze spínačů představuje data, která budou naprogramována do SRAM a EEPROM.
- **SRAMDATA** , **EPROMDATA** – piny Pmod konektorů, slouží pro obousměrnou komunikaci s datovými piny SRAM/EPROM paměti.
- **JA, JB, JC, JD** – piny Pmod konektorů komunikující s adresovými vstupy SRAM/EPROM.
- **Data\_io** – datové piny SRAM paměti na desce NEXYS3.

### 7.1.2. Výstupní porty

- **SI** – „*serial input*“, pin Pmod konektoru, zprostředkovává komunikaci se vstupem EEPROM paměti.
- **CS** – „*chip select*“, pin Pmod konektoru, slouží k řízení komunikace s EEPROM. Nastavením do log. ‚1‘ je znemožněna komunikace, naopak povolení komunikace odpovídá log. ‚0‘.
- **SCK** – pin Pmod konektoru, poskytuje EEPROM hodinový signál.
- **R\_W** – pin Pmod konektoru, slouží k rozlišení zápisu a čtení SRAM.
- **Tx** – pin Usb-RS232 rozhraní, slouží k odesílání dat z desky do počítače.
- **addr\_o, clk\_o, addr\_valid\_o, cntl\_reg\_enable\_o, chip\_enable\_o, output\_enable\_o, write\_en\_o, lower\_byte\_en\_o, upper\_byte\_en\_o** – jedná se o piny SRAM paměti, význam je popsán v kapitole 7.3

### 7.1.3. Popis funkce

Hlavní funkcí DP2 je propojení jednotlivých komponent projektu mezi sebou, čehož se docílí pomocí **instanciace komponent** – signály deklarované v komponentě se propojí se signály architektury.

- **MAINCLOCK 1MHz**

Krystalový oscilátor na desce vytváří **hodinový signál 100MHz**, ale vzhledem k použitým paměťovým součástkám, především EEPROM, která je schopna pracovat s **maximální frekvencí 1MHz**, bylo třeba taktovací frekvenci podstatně snížit. K tomu byl použit DCM, který nejprve z 100MHz signálu vytvoří 16MHz signál, ten je poté pomocí děličky šestnácti upraven na výsledný signál 1MHz. Se signálem 1MHz pod názvem mainclock pracují i všechny ostatní komponenty projektu.

### Část VHDL kódu – vznik mainclock

```
process (clk10MHZ)
variable divider : std_logic_vector (3 downto 0);
begin
    if clk10MHZ='1' and clk10MHZ'event then
        divider:=divider+1;
        mainclock <= divider(3);
    end if;
end process;
```

- **MASTER\_ADDRESS**

Jak již bylo zmíněno v předchozí kapitole, paměti SRAM a EPROM sdílejí adresovou sběrnici. Proto byl vytvořen pouze **jeden** hlavní adresovací signál **master\_address** o velikosti 13b, společný pro všechny paměti.

Jelikož se pracuje s pamětmi s paralelním i sériovým rozhraním, je zřejmé, že **zápis** nebo **čtení** jediné adresy bude pro každou součástku jinak časově náročný. Z toho vyplývá, že doba, kdy jsou paměti schopny inkrementace adresy, bude **rozdílná**. Proto je daný problém v DP2 ošetřen níže uvedenou **podmínkou** – inkrementace hlavní adresy bude povolena, pouze pokud jsou **všechny** paměti **připraveny** inkrementovat.

```
enable_master_address    <=    enable_address_SRAM    and
enable_address_EEPROM and enable_address_EEPROM;
```

Hlavní adresa se používá pro adresování pamětí při **programování**, ale také při **kontrole**. Z toho důvodu je nutné zajistit **reset adresy** po dokončení programování nebo po každém kole kontroly. Jedním ze způsobů jak lze resetovat hlavní adresu je použití tlačítka reset. Nebo se vynuluje automaticky, pokud jsou všechny stavové automaty ostatních komponent ve výchozím stavu.

```
reset_master_address    <=    reset    or    not(state_program    or
state_check or state_reading);
```

Přestože paměti sdílejí adresové vodiče, každý vodič znamená pro SRAM a EPROM jinou pozici bitu v adrese, proto se musí adresa EPROM **přeorganizovat** do správného pořadí. Záznam správné adresy je důležitý při následné analýze výsledků.

```
LOG_ADDR_EEPROM(0) <= LOG_ADDR_EEPROM_spatne(10);
LOG_ADDR_EEPROM(1) <= LOG_ADDR_EEPROM_spatne(11);
LOG_ADDR_EEPROM(2) <= LOG_ADDR_EEPROM_spatne(2); ...
```

Signál `master_address` je veden přímo na piny Pmod konektorů.

```
JB(0) <= master_address(0);
JB(5) <= master_address(1);
JA(3) <= master_address(2);...
```

#### Část VHDL kódu – reset a inkrementace `master_address`:

```
process (reset_master_address, mainclock)
begin
    if reset_master_address = '1' then
        master_address <= (others => '0');
    else
        if mainclock='1' and mainclock'event then
            if enable_master_address = '1' then
                master_address <= master_address + 1;
            end if;
        end if;
    end if;
end process;
```

- **DISPLAY, LED**

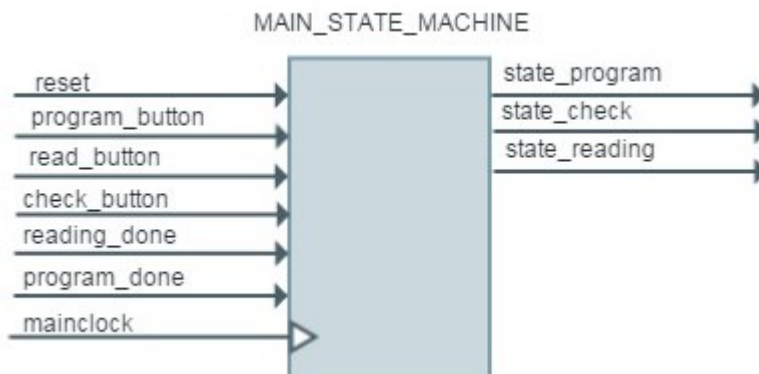
Led a displej desky, byly využity pro **indikaci**, a tedy zpětnou vazbu, probíhajících procesů. **Led** (7:4) slouží k indikaci **stavu** programování či kontroly paměti EEPROM a led (3:1) náleží SRAM.

```
led (3 downto 1) <= stav1(3 downto 1);
led (7 downto 4) <= stav2;
```

Vzhledem k tomu, že je celá tato práce koncipována jako výzkum, není jasné, zda vůbec dojde k nějakým změnám během ozařování pamětí. Na **displeji** desky proto bude během měření zobrazena **aktuální adresa paměti výsledků**, která může zůstat nezměněná, nebo její zvyšující se hodnota bude indikovat přibývajících záznamů změn. Což umožní již během měření odhadnout výsledek pokusu. Adresa paměti výsledků bude na displeji zobrazena také během odesílání dat do počítače. Pokud by byl objem dat příliš velký, odesílání přes sériovou linku by mohlo trvat delší dobu. Údaje na displeji tedy mohou sloužit i jako indikace stavu odesílání a tím lze např. předejít předčasnému ukončení odesílání uživatelem a ztrátě některých dat.

```
an <= not anode;
seg (6 downto 0) <= not cathode;
addr_o <= adresa_CRAM;
```

## 8.2 main\_state\_machine (MSM)



Obrázek 27 Entita main\_state\_machine

### 7.2.1. Vstupní porty

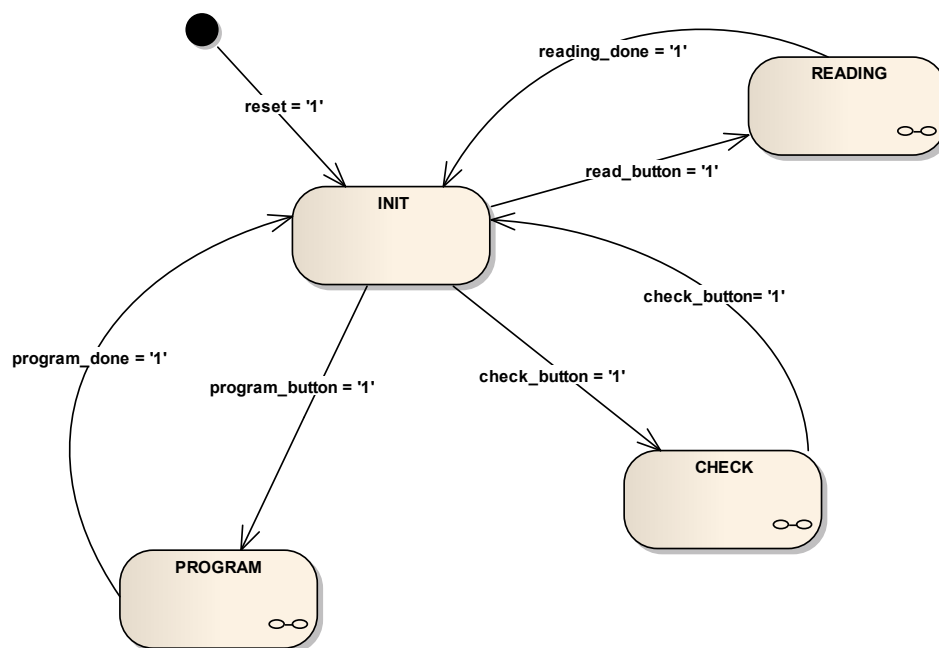
- **Reset** – po stisknutí tlačítka reset, je stavový automat MSM uveden do výchozího stavu a vyčkává na stisk jednoho z tlačítek: programování, čtení, kontrola.
- **Program\_button, read\_button, check\_button** – signály nabývají hodnoty log. ,1‘, pouze pokud bylo stisknuto odpovídající tlačítko, jinak mají hodnotu log. ,0‘.
- **Reading\_done** – signál do log. ,1‘ vystavuje komponenta „LOG“, signalizuje dokončení odesílání dat přes sériovou linku.
- **Program\_done** - signál do log. ,1‘ vystavuje komponenta „programování“, signalizuje dokončení programování všech pamětí.

### 7.2.2. Výstupní porty

- **State\_program, state\_check, state\_reading** – signál má hodnotu log. ,1‘ po stisku odpovídajícího tlačítka a uvádí do chodu navazující FSM řídící programování, kontrolu nebo aktivuje odesílání dat přes sériovou linku.

MSM lze popsat jako **přepínač** mezi třemi základními **funkcemi** celého návrhu: programování, kontrola dat, odesílání dat přes sériovou linku.

Ke konstrukci stavového registru je použit signál **výčtového typu**, jehož hodnoty odpovídají množině všech možných stavů FSM: **st1\_init, st2\_program, st3\_check, st4\_reading**. Stavy 2-3 představují další samostatné FSM, které budou podrobně rozebrány v následujících kapitolách. Činnost MSM ilustruje stavový diagram **Obrázek 28**.

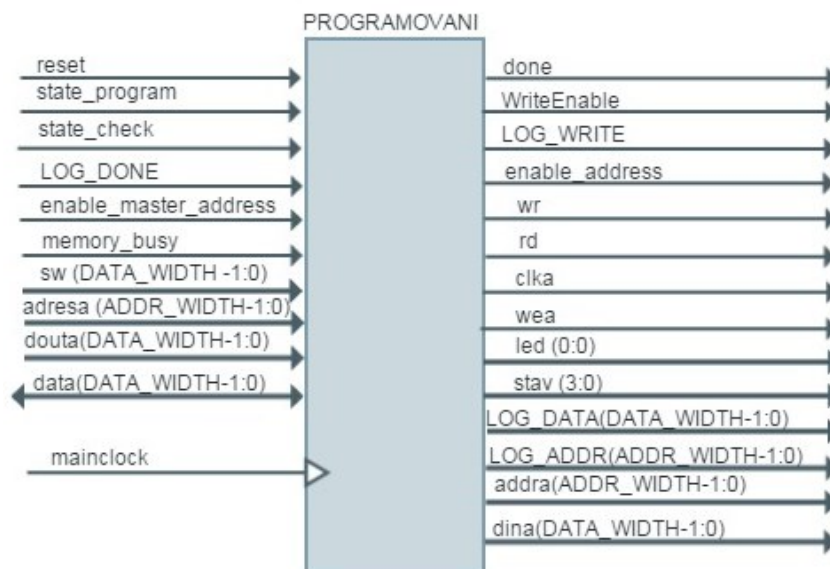


Obrázek 28 Stavový diagram MSM

### 7.2.3. Stavy

- **St1\_init** – MSM přechází do výchozího stavu st1 na základě stisku tlačítka reset, nebo jakmile se dokončí jeden z procesů programování, kontrola, odesílání dat. Pokud se FSM nachází ve stavu st1, jsou všechny výstupní signály nastaveny na hodnotu log. ,0‘.
- **St2\_program** – ze stavového diagramu je zřejmé, že do st2 přechází MSM po stisku tlačítka programování. Signál state\_program, který je spouštěcím signálem FSM „programování“ (viz následující kapitola), přechází do hodnoty log. ,1‘. MSM setrvává ve stavu st2 dokud není programování dokončeno a „programování“ nevystaví signál program\_done do log. ,1‘.
- **St3\_check** – do st2 přechází MSM po stisku tlačítka kontrola, ale na rozdíl od předchozího případu, je doba setrvání v st3 záležitostí uživatele, tedy do výchozího stavu je MSM uveden až opětovným stiskem tlačítka kontrola. Opět platí, pouze signál state\_check, který je také spouštěcím signálem FSM „programování“, přechází do hodnoty log. ,1‘, je spuštěna kontrola a ukládání dat o zjištěných chybách.
- **St4\_reading** – stiskem tlačítka odesílání se nastaví state\_reading do log. ,1‘ a je aktivován FSM „LOG“, začne vyčítání uložených záznamů a odesílání dat přes sériovou linku. Jakmile je proces odesílání dokončen, „LOG“ vystaví signál reading\_done do log. ,1‘ a MSM je opět uveden do výchozího stavu.

## 8.3 programovani



Obrázek 29 Entita programovani

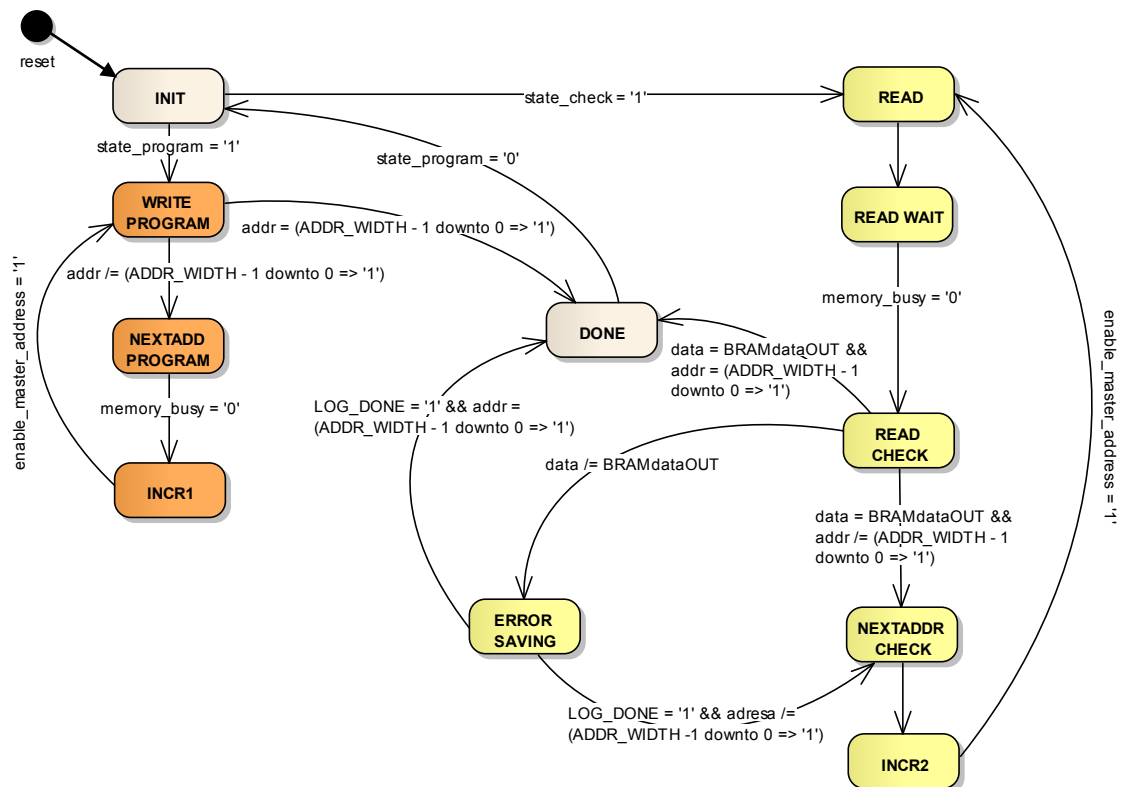
### 7.3.1. Vstupní, obousměrné porty

- **Reset** – uvádí stavový automat do výchozího stavu.
- **State\_program, state\_check** – signály z MSM, pokud mají hodnotu log. „1“ spouštějí odpovídající akci programování nebo kontrola, viz **Obrázek 30**.
- **Sw** – signály ze spínačů na desce NEXYS3, jejich kombinace představuje data, která budou naprogramována do SRAM a EEPROM, a také jako referenční data do všech BRAM.
- **Adresa** – jedná se o odpovídající počet bitů master\_address z DP2 – každá paměť má jiný počet adresových vodičů. Využívá se při počátečním programování, při kontrole, ale také pro adresování BRAM.
- **Enable\_master\_address** – signál z DP2 povolující inkrementaci hlavní adresy.
- **Douta** – data uložená v BRAM odpovídací paměti, slouží k porovnání dat v předchozím kole kontroly s aktuálním obsahem paměti.
- **Data** – SRAMDATA, EPROMDATA, EEPROMDATA – data, která jsou posílána do paměti při programování nebo data čtená z paměti při kontrole. Tato sběrnice se může nacházet ve stavu vysoké impedance.

- **Memory\_busy** – signál je využit pouze při programování a čtení EEPROM, protože má sériové rozhraní a R/W operace probíhají mnohem pomaleji než u paměti s paralelním rozhraním. Proto se memory\_busy nastaví do log. ,1‘ během ukládání nebo čtení dat, čímž je ošetřeno, že hlavní adresa nebude inkrementována, dokud EEPROM nedokončí správně zápis nebo čtení. Pro zbývající paměti je signál trvale na hodnotě log. ,0‘.
- **LOG\_DONE** – signál indikující dokončení ukládání chyby komponentou „LOG“.

### 7.3.2. Výstupní porty

- **Done** – signál indikující dokončení programování odpovídající paměti.
- **WriteEnable** – signál povolující zápis do SRAM a EEPROM jen během programování.
- **Enable\_address** – pokud nabývá hodnotu log. ,1‘ znamená, že odpovídající paměť dokončila všechny operace a je připravena inkrementovat adresu.
- **Wr, Rd** – signály pro „Programovani\_EEPROM“, EEPROM se odešle instrukce pro zápis/čtení. Ostatní paměti signály nevyužívají.
- **Clka, wea, addra, dina** – signály pro BRAM, hodinový signál - mainclock, signál povolující zápis do BRAM, adresa, vstupní data.
- **LOG\_WRITE** – nastaví se do log. ,1‘ pokud byla detekována chyba, signál pro komponentu „LOG“ k započetí ukládání dat o změně.
- **LOG\_DATA, LOG\_ADDR** – data pro komponentu „LOG“ – jedná se o adresu buňky, kde byla zjištěna změna při kontrole a změněná data. Tato data budou spolu s typem paměti a časem detekce chyby tvořit datový paket, který se ukládá do CRAM – paměti výsledků.



Obrázek 30 Stavový diagram „programování“

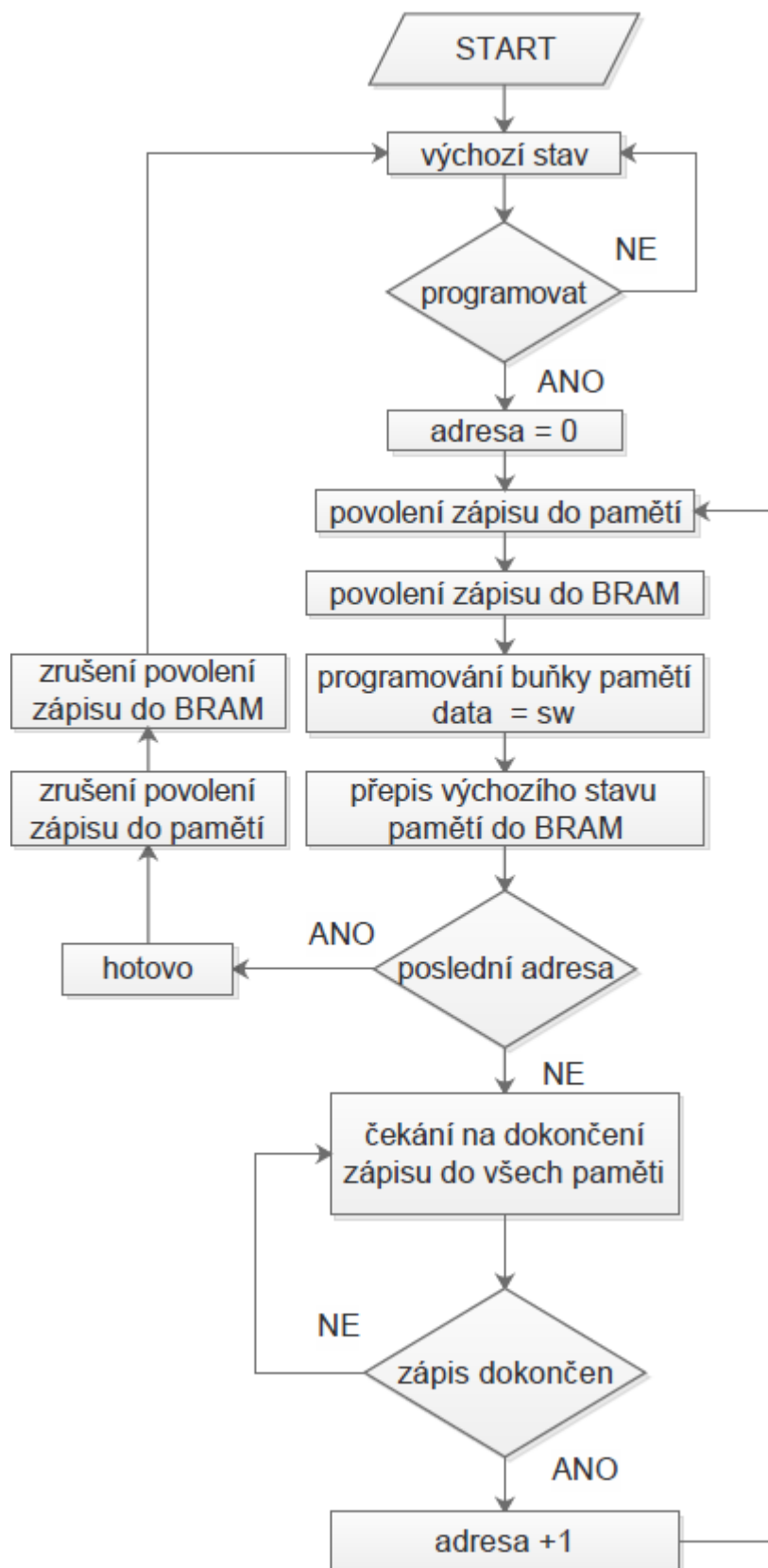
FSM *programování* je navržen tak, aby byl **univerzální** pro všechny tři paměti. Rozdíly v šířkách adresových a datových sběrnic jsou řešeny pomocí deklarace **generics**, které se v instanciaci komponenty nastaví na konkrétní šířky, stejně tak jako se vstupní a výstupní porty propojí se signály z následujících komponent: *LOG*, *MSM*, *programování\_EEPROM*. FSM realizuje jak **programování** pamětí (oranžová větev stavového diagramu), tak následnou **kontrolu** (žlutá větev stavové diagramu).

Ke konstrukci stavového registru je použit signál **výčtového typu**, jehož hodnoty odpovídají množině všech možných stavů FSM: **st1\_init**, **st2\_nextaddr\_program**, **st3\_write\_program**, **st4\_done**, **st5\_nextaddr\_check**, **st6\_read**, **st7\_error\_saving**, **st8\_incr1**, **st9\_incr2**, **st10\_read\_wait**, **st11\_read\_check**.

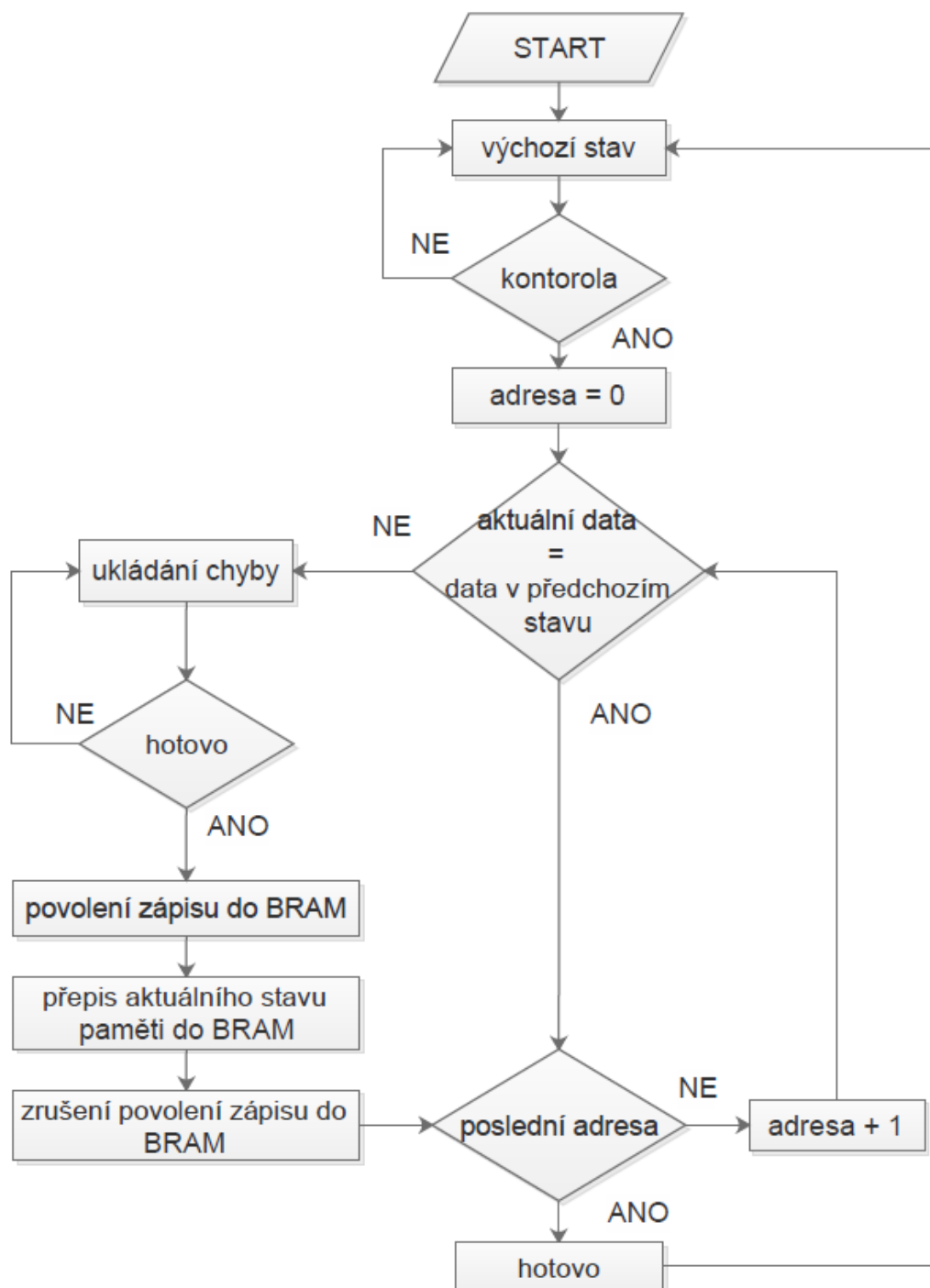


### 7.3.3. Stavy

- **St1\_init** - *programování* přechází do výchozího stavu st1 na základě stisku tlačítka reset, jakmile je dokončeno programování celé paměti nebo už byl zkontrolován celý obsah paměti.
- **St3\_write\_program** – stiskem tlačítka programování se povolí zápis do konkrétní paměti a od nulté adresy se začíná programovat daty – kombinace sw. Zároveň je povolen zápis do BRAM, která je rovněž naplněna daty ze sw. Dokud není naprogramována i poslední buňka paměti, přechází se do čekacího stavu st2. Ale když je programování celé paměti dokončeno, FSM je uveden do výchozího stavu.
- **st2\_nextaddr\_program** – st2 představuje čekání na dokončení všech zapisovacích procesů paměti EEPROM, teprve poté může proběhnout inkrementace adresy.
- **st8\_incr1** – signál enable\_address odpovídající paměti je nastaven do log. ,1‘, což znamená, že je připraven inkrementovat adresu. Až tento signál vystaví do log. ,1‘ i ostatní paměti, může se hlavní adresa zvětšit o jedničku. FSM přejde opět do stavu st3 a celý proces se opakuje.
- **st6\_read** – pin R\_W paměti SRAM je nastaven do log. ,1‘, paměti EEPROM se odešle instrukce čtení.
- **st10\_read\_wait** – obdobně jako u st2 – je třeba vyčkat, dokud EEPROM nepracuje instrukci čtení a nevystaví data na výstupním pinu.
- **st11\_read\_check** – nyní je možné porovnávat obsah paměti v předchozím kroku s aktuálním obsahem. Jestliže nedojde k žádné změně a zatím nebyla překontrolována poslední adresa, přejde FSM do stavu čekání na povolení inkrementace adresy, ale pokud už byla zkontrolována i poslední adresa, přechází se do výchozího stavu. Pokud se však zaznamená změna, přechází se k ukládání chyby (viz 8.6), zároveň je povolen zápis do BRAM, kde se uloží změněná data. Je-li chyba uložena, může FSM rovněž přejít do čekání na povolení inkrementace, nebo do výchozího stavu. Dokud uživatel kontrolu neukončí, celý proces bude neustále probíhat opakovaně.
- **st9\_incr2** – st9 realizuje naprosto stejné funkce jako st8. Po inkrementaci adresy se znovu přechází do st6 a proces se znovu opakuje.

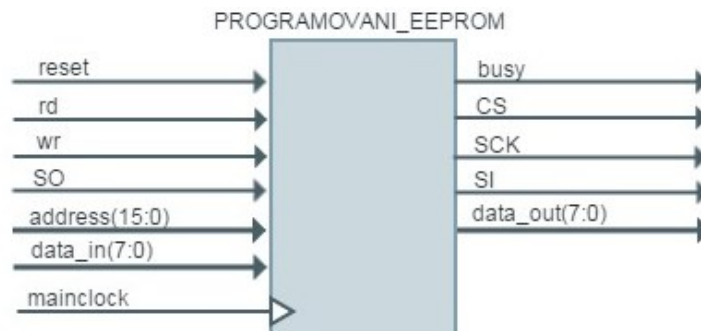


Obrázek 31 Princip programování



Obrázek 32 Princip kontroly

## 8.4 Programovani\_EEPROM



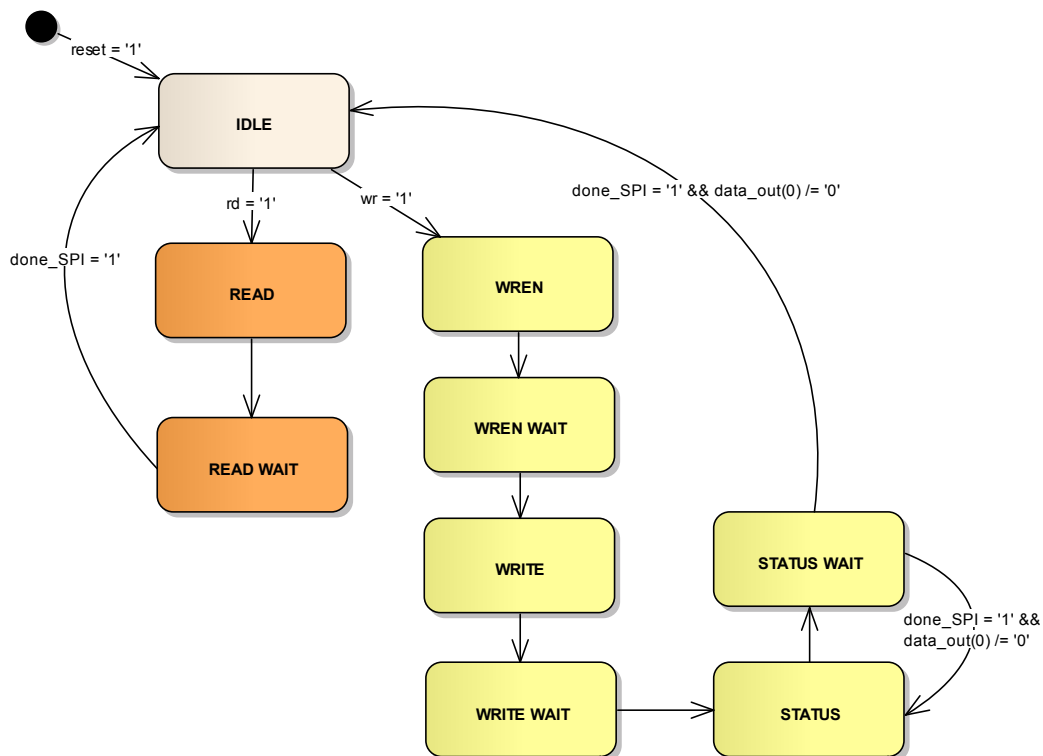
Obrázek 33 Entita Programování EEPROM

### 7.4.1. Vstupní porty

- **Reset** – uvádí stavový automat do výchozího stavu.
- **Rd** – signál vystavuje FSM „programovani“, pokud je v log. ,1‘ spustí proces vyčítání z paměti.
- **Wr** – signál vystavuje FSM „programovani“, pokud je v log. ,1‘ spustí proces zápisu do paměti.
- **SO** – výstupní pin EEPROM.
- **Address** – adresa EEPROM je široká 16b a první tři MSB jsou *don't care*, master\_addr má pouze 13b, adresová sběrnice Address má tedy následující obsah:  
`"000" & master_address(12 downto 0);`
- **Data\_in** – data posílána paměti. Obsah data\_in se liší podle právě probíhajícího procesu.

### 7.4.2. Výstupní porty

- **Busy** – pokud je v log. ,1‘ signalizuje zaneprázdněnost paměti probíhající operací.
- **CS** – chip select, řídí komunikaci s EEPROM. Komunikace je možná, pouze pokud je v log. ,0‘.
- **SCK** – hodinový signál.
- **SI** – vstupní pin EEPROM.
- **Data\_out** – data z EEPROM – budou se číst buď data uložená v paměti, nebo obsah stavového registru.



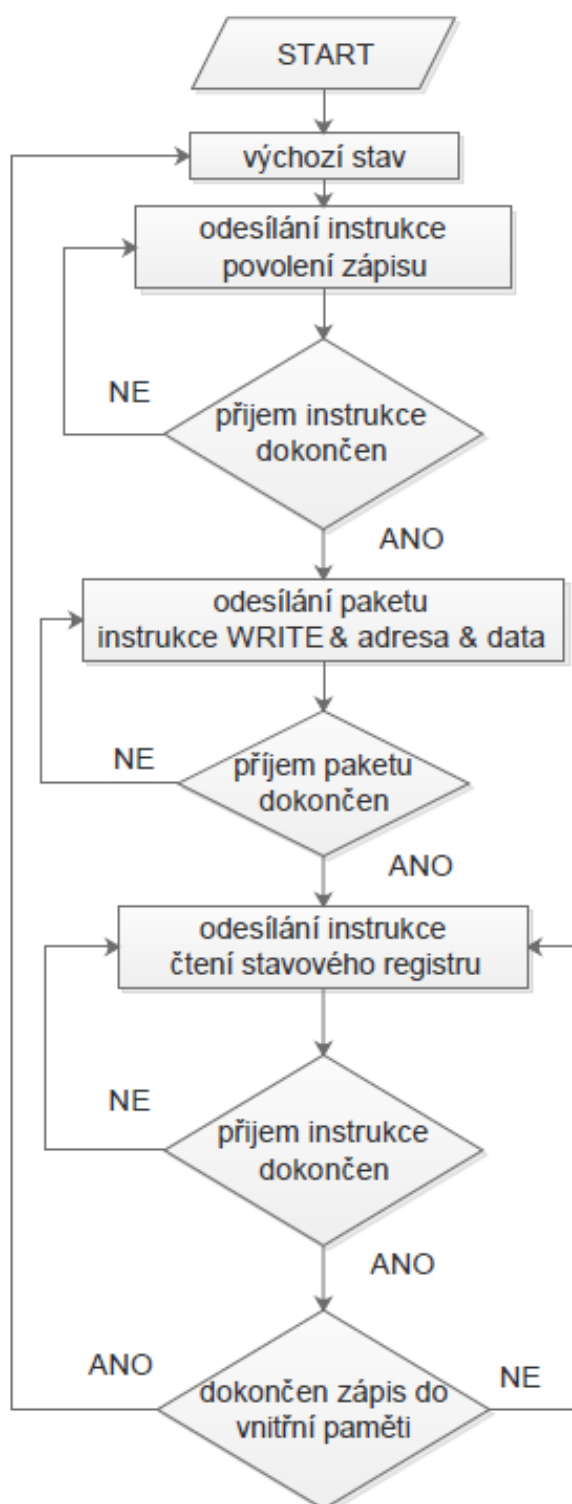
Obrázek 34 Stavový diagram

FSM je pomocí komponenty „*SPI\_controller*“ (viz. dále) schopen komunikovat se sériovým rozhraním paměti EEPROM. Komunikace začíná nastavením CS pinu na log. ‚0‘ a poté odesláním jedné z **instrukcí**: povolení zápisu (WREN), zápis (WRITE), čtení (READ) a čtení stavového registru (RDSR) a končí nastavením CS pinu do log. ‚1‘ po určitém počtu taktů SCK. Ke kompletnímu odeslání zapisovací sekvence, tedy instrukce/adresy/dat, je potřeba držet CS pin v log. ‚0‘ **32 taktů SCK**, stejně tak jako při čtecí sekvenci. Zato k sekvenci čtení stavového registru, tedy odeslání instrukce a vystavení dat na SO pinu, musí být CS pin v log. ‚0‘ **16 taktů SCK** a při sekvenci povolení zápisu, kdy se odesílá pouze instrukce a nečeká se na žádnou odpověď paměti, postačí **8 taktů SCK**.

Ke konstrukci stavového registru je použit signál **výčtového typu**, jehož hodnoty odpovídají množině všech možných stavů FSM: **st1\_idle**, **st2\_read**, **st3\_read\_wait**, **st4\_wren**, **st5\_wren\_wait**, **st6\_write**, **st7\_write\_wait**, **st8\_status**, **st9\_status\_wait**.

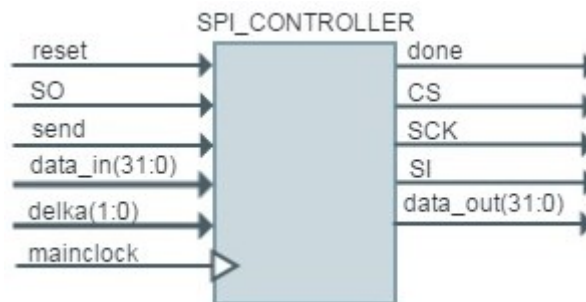
### 7.4.3. Stavy

- **St1\_idle** – stejně jako v předchozích případech uvádí FSM do výchozího stavu signál reset. Do st1 se FSM vrací po dokončení čtecí sekvence nebo pokud data ze stavového registru indikují dokončení zápisu. Pouze v tomto stavu je signál busy na log. ,0‘.
- **St2\_read** – komponentě „*SPI\_controller*“ se odesílá příkaz k započetí komunikace, spolu s počtem taktů nutných pro READ sekvenci, čtecí instrukce a adresa.
- **st3\_read\_wait** - „*SPI\_controller*“ pomocí posuvného registru odesílá jednotlivé bity paketu na SI pin. Když jsou data vystaveny na SO pinu, obdobně za použití posuvného registru ukládá, jednotlivé bity. Po skončení posuvu vystaví „*SPI\_controller*“ signál done do log. ,1‘ a FSM přechází do výchozího stavu.
- **st4\_wren** - komponentě „*SPI\_controller*“ se odesílá příkaz k započetí komunikace, spolu s počtem taktů nutných pro WREN sekvenci a WREN instrukce. Průběh zápisu ilustruje stavový diagram **Obrázek 35**
- **st5\_wren\_wait** - „*SPI\_controller*“ pomocí posuvného registru odesílá jednotlivé bity paketu na SI pin. Po odeslání instrukce se signál done, nastaví na log. ,1‘, FSM může přejít do dalšího stavu.
- **st6\_write** - komponentě „*SPI\_controller*“ se odesílá příkaz k započetí komunikace, spolu s počtem taktů nutných pro WRITE sekvenci, WRITE instrukce, adresa a data.
- **st7\_write\_wait** – vystavení done do log. ,1‘ znamená, že je dokončeno posouvání celého paketu na SI pin.
- **st8\_status** - komponentě „*SPI\_controller*“ se odesílá příkaz k započetí komunikace, spolu s počtem taktů nutných pro RDSR sekvenci, RDSR instrukce.
- **st9\_status\_wait** - „*SPI\_controller*“ pomocí posuvného registru odesílá jednotlivé bity paketu na SI pin. A poté vysouvá odpověď paměti. Kontrola stavového registru je důležitá proto, že paměť spustí zapisovací cyklus teprve až potom, co přijme vše 32b paketu. Pokud by byl CS nastaven do log. ,1‘ dříve, data by nebyla správně uložena. Proto se teprve až je LSB stavového registru v log. ,0‘(paměť kompletně dokončila zápis) může přejít do výchozího stavu st1.



**Obrázek 35** Stavový diagram - zápis do EEPROM

## 8.5 SPI\_Controller



Obrázek 36 Entita SPI\_Controller

SPI řadič na základě instrukcí z komponenty „*Programovani\_EEPROM*“ realizuje odesílání a přijímání dat ze sériových pinů EEPROM. Jeho další nezbytnou funkcí je povolování sériových hodin pro komunikaci a řízení log. hodnoty CS pinu. Vystavením signálu done do log. ,1‘ pak indikuje ukončení své činnosti pro „*Programovani\_EEPROM*“.

- **Dekódování délky sekvence**

Délka sekvence určuje horní hodnotu čítače. Dokud čítač nepřeteče, jsou povoleny SCK.

**Část VHDL kódu:**

```
case delka is
  when "00" => counter_top <= "001000";
  when "01" => counter_top <= "010000";
  when "10" => counter_top <= "011000";
  when "11" => counter_top <= "100000";
  when others => counter_top <= "100000";
end case;
```

- **Posuvný registr**

Po vystavení signálu send do log. ,1‘ komponentou „*Programovani\_EEPROM*“ se započne komunikace nastavením CS do log. ,0‘. Data\_in se uloží do registru a když je povoleno SCK, začne odesílání dat na SI pin posuvným registrem. Zároveň se místo po vysunutých datech plní daty z SO pinu. Jakmile je celý proces posouvání dokončen, obsah registru je uložen do signálu data\_out, zruší se povolení SCK a CS jde do log. ,1.



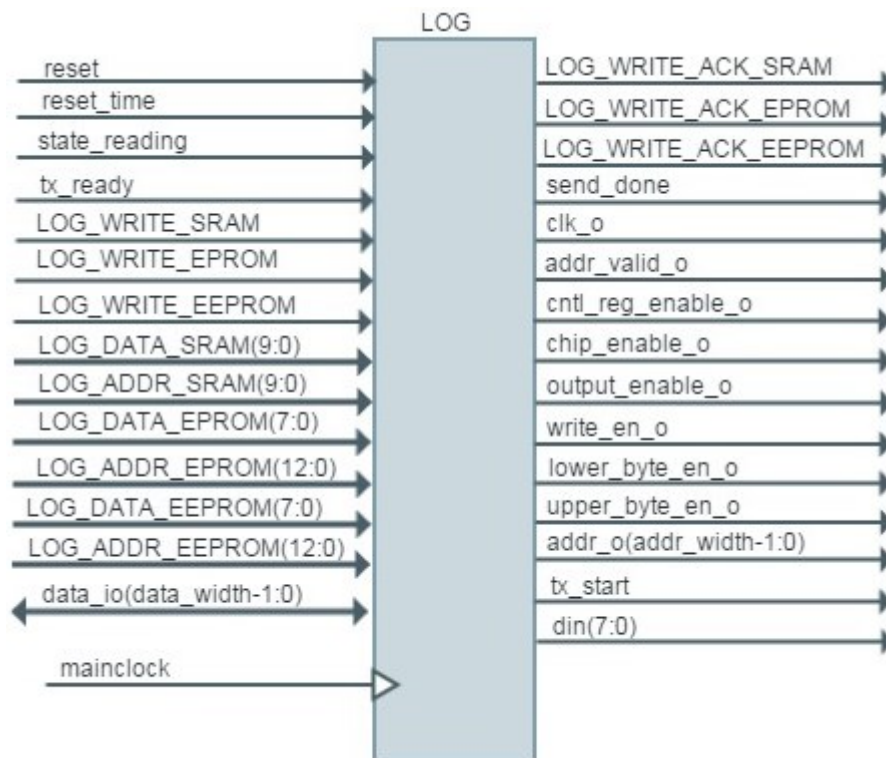
### Část VHDL kódu – posuvný registr:

```

if chip_select = '0' then
    clock_enable <='1';
end if;
if clock_enable = '1' then
    if SerialClock = '1' then
        counter <= counter+1;
        SI <= shift_reg(31);
        for i in 0 to 32-2 loop
            shift_reg(i+1) <= shift_reg(i);
        end loop;
        shift_reg(0) <= SO;
    end if;
    SerialClock <= not SerialClock;
else
    SerialClock <='0';
end if;

```

## 8.6 LOG



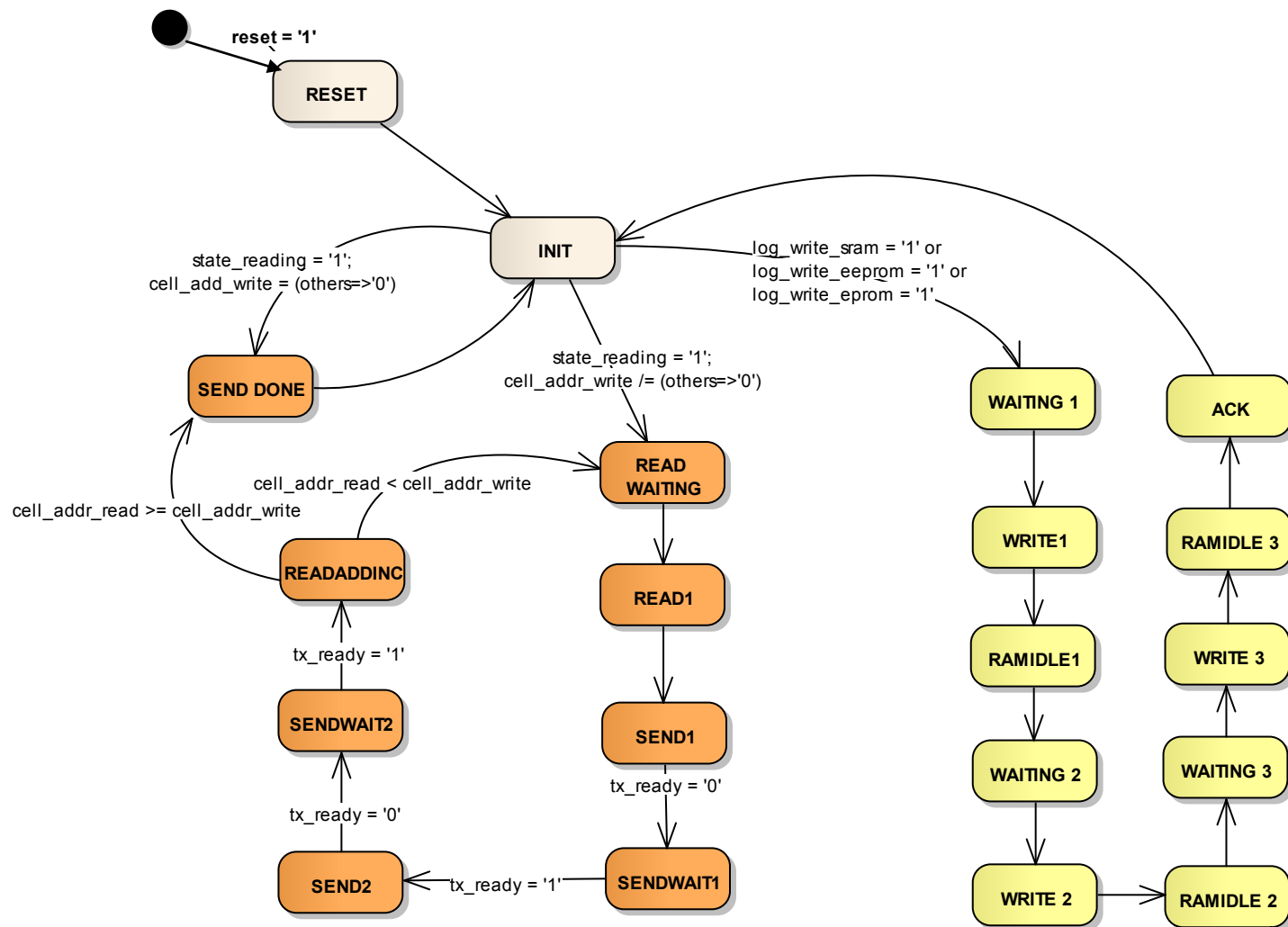
Obrázek 37 Entita LOG

### 7.6.1. Vstupní porty

- **Reset** – signál resetuje časovač záznamu a uvádí FSM do stavu reset.
- **Reset\_time** – signál z DP2, resetuje časovač vždy, když je state\_check v log. ,0‘
- **State\_reading** – stiskem tlačítka odesílání se nastaví do log ,1‘ a tím se spustí vyčítání dat z paměti výsledku – CRAM a odesílání do počítače.
- **LOG\_WRITE\_XXX** – signál z komponenty „programování“, je v log. ,1‘ pokud byla detekována chyba v jakékoli paměti a znamená, že „LOG“ začne ukládat záznam o změně.
- **LOG\_ADDR\_XXX** – adresa buňky, ve které byla detekována změna.
- **LOG\_DATA\_XXX** – změněná data, která se budou ukládat do paměti výsledků
- **Data\_io** – data ukládána a vyčítána z CRAM.
- **Tx\_ready** – signál z komponenty „transmitter“, log. ,1‘ indikuje, připravenost vysílače k vysílání.

### 7.6.2. Výstupní porty

- **LOG\_WRITE\_ACK\_XXX** – pokud je signál v log. ,1‘, indikuje pro komponentu „programování“ ukončení ukládání záznamu.
- **Tx\_start** – nastavením na log. ,1‘ dává komponentě „transmitter“ povolení k započetí odesílání dat přes sériovou linku do počítače.
- **Send\_done** – signál se nastaví do log. ,1‘ pokud byla vyčtena a odeslána všechna data uložená v paměti výsledků, což znamená pro MSM ukončení procesu odesílání dat.
- **Din** – data, která se budou odesílat do počítače.
- **addr\_o, clk\_o, addr\_valid\_o, cntl\_reg\_enable\_o, chip\_enable\_o, output\_enable\_o, write\_en\_o, lower\_byte\_en\_o, upper\_byte\_en\_o** – jedná se o piny CRAM paměti, význam je popsán v kapitole 7.3



Obrázek 38 Stavový diagram „LOG“

„LOG“ realizuje ukládání výsledků do CRAM, jejich následné vyčítání a pomocí komponenty „transmitter“ odesílání přes sériovou linku do počítače. Záznam o změně dat se ukládá ve tvaru 48b **paketu – log\_entry**, který obsahuje: **typ paměti** (3b), **časový záznam** (24b), **adresu** (16/12/10b) a **změněná data** (8/4b). Jak již bylo zmíněno v 7.3 do CRAM lze najednou uložit vždy jen 2B (UB, LB), má tedy celkem 16 datových vstupů, proto musí být ukládání do CRAM rozděleno na tři zapisovací cykly. CRAM se již neadresuje hlavní adresou, při zápisu se adresa inkrementuje po každém zapisovacím cyklu. Vyčítání a odesílání dat do počítače se pak také realizuje pro UB a LB zvlášť a adresa je inkrementována až po odeslání obou bytů.

Ke konstrukci stavového registru je použit signál **výčtového typu**, jehož hodnoty odpovídají množině všech možných stavů FSM: **st1\_reset**, **st2\_init**, **st4\_write1**, **st5\_write2**, **st6\_write3**, **st7\_ack**, **st8\_waiting1**, **st9\_waiting2**, **st10\_waiting3**, **st11\_readwaiting**, **st12\_read1**, **st13\_send1**, **st14\_send\_done**, **st15\_sendwait1**, **st16\_send2**, **st17\_sendwait2**, **st21\_readaddinc**, **st18\_ramidle1**, **st19\_ramidle2**, **st20\_ramidle3**.

### 7.6.3. Stavy

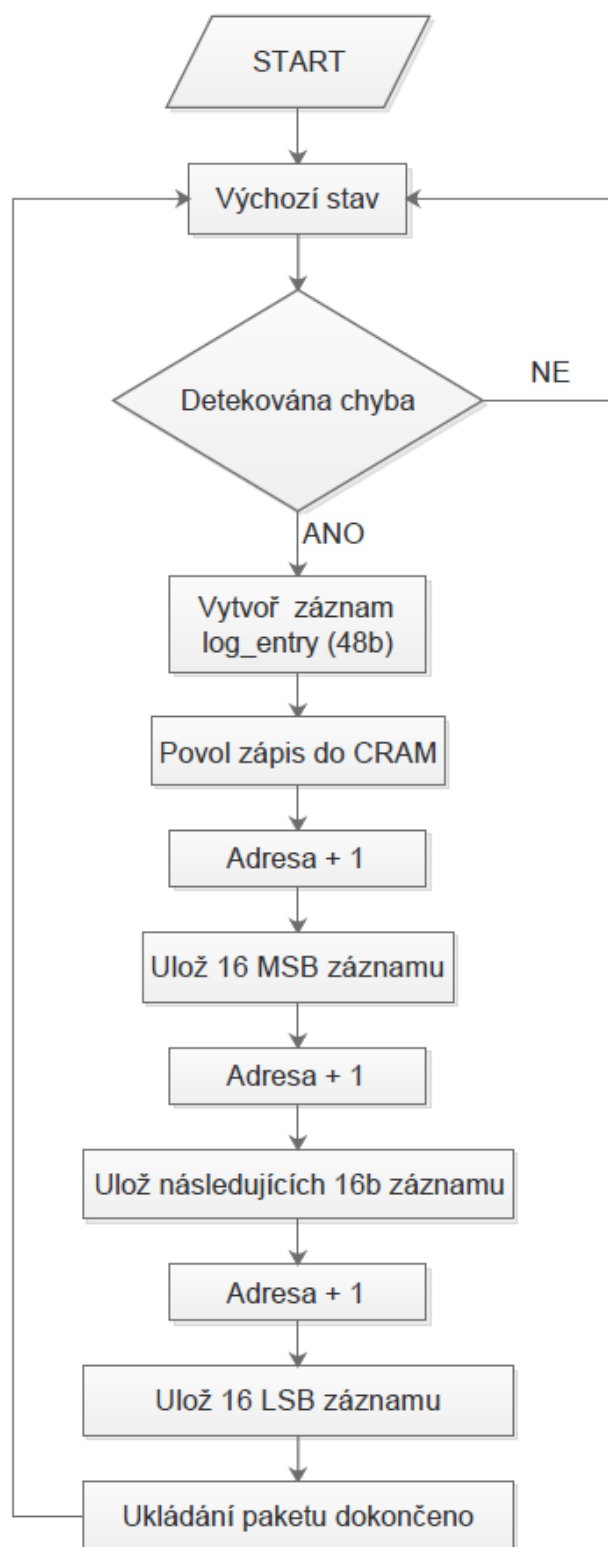
- **St1\_reset** – do st1 přechází FSM po stisku tlačítka reset, čtecí i zapisovací adresa je resetována, stejně jako obsah **log\_entry**. FSM poté rovnou přechází do st2.
- **St2\_init** – FSM setrvává v st2 tak dlouho než „programování“ detekuje chybu nebo pokud není stisknuto tlačítko odesílání. Pokud není v CRAM nic k vyčtení, přechází se do stavu st14, pro MSM je vystaven signál **send\_done** do log. ,1‘ signalizující dokončení odesílání dat. Zpět do st2 se FSM vrátí až dokončí ukládání záznamu o změně, nebo až se dokončí odesílání všech dat uložených v CRAM.

#### Ukládání záznamu

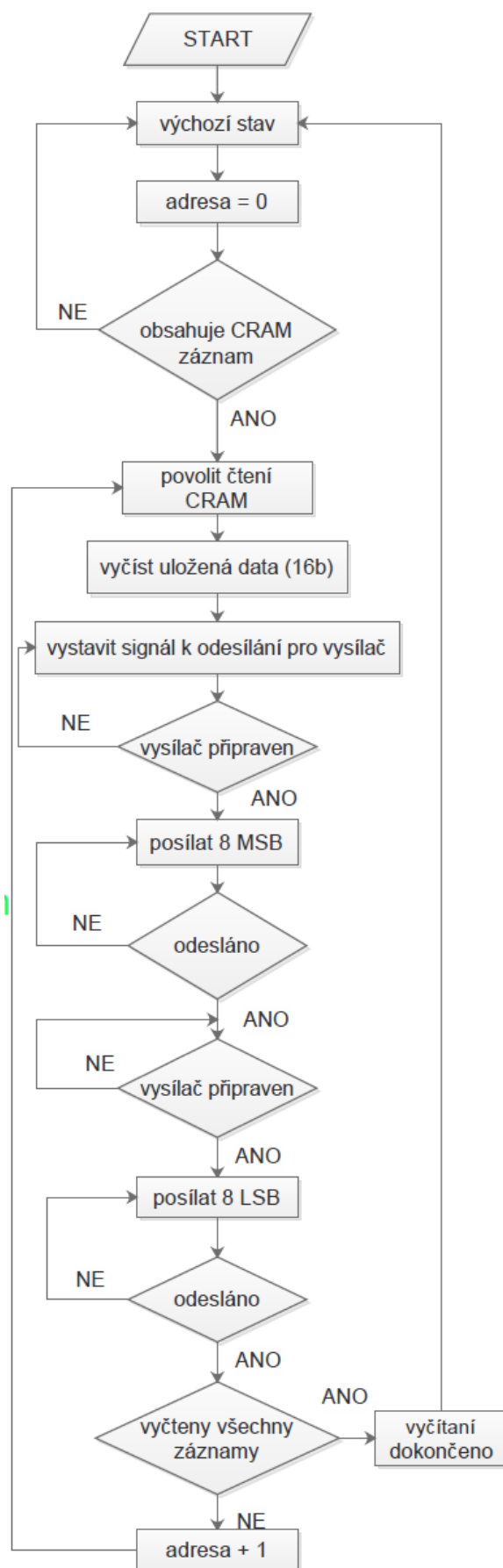
- **St8\_waiting1**, **st9\_waiting2**, **st10\_waiting3** – v těchto stavech je povolena komunikace s CRAM (CE, UB, LB v log. ,0‘) a do signálu **data\_i** je postupně od MSB formováno 16b záznamu o chybě.
- **St4\_write1**, **st5\_write2**, **st6\_write3** – nyní je povolen zápis do CRAM - **write\_en\_o** má hodnotu log. ,0‘ – a je inkrementována zapisovací adresa.
- **st18\_ramidle1**, **st19\_ramidle2**, **st20\_ramidle3** – v těchto stavech se vždy zruší povolení komunikace s CRAM.
- **st7\_ack** – může se stát, že dojde ke změně ve více pamětech, proto se musí počkat, dokud není signál **LOG\_WRITE\_ACK\_XXX** nastaven na hodnotu log. ,1‘ a potom se teprve může ukládat další záznam, případně kontrolovat další buňka.

### Čtení CRAM a odesílání dat přes sériovou linku:

- **st11\_readwaiting** – FSM přechází do st11 pokud je stisknuto tlačítko odesílání a pokud je v CRAM uložen nějaký záznam (`cell_addr_write /= (others=>'0')`), povolí se komunikace s CRAM a výstup – signál `output_enable` se nastaví do log. ,0'. Zpět do st11 se FSM vrací za podmínky, že ještě nebyl vyčten a odeslán celý obsah CRAM.
- **st12\_read1** – z CRAM jsou vyčtena data a uložena do pomocné proměnné `data_to_send`.
- **st13\_send1, st16\_send2** – povolení komunikace s CRAM i povolení výstupu je zrušeno. V každém stavu je na datový vstup komponenty „*transmitter*“ vloženo 8b, začíná se od MSB a komponentě „*transmitter*“ je vystaven signál povolující start přenosu přes sériovou linku - `tx_start` je nastaven do log. ,1'. Až bude vysílač připraven k přenosu – signál `tx_ready` v log. ,0' – může FSM přejít do dalšího stavu.
- **st15\_sendwait1, st17\_sendwait2** – FSM čeká, dokud není odesláno všech 8b dat. Když je odesílání dokončeno, signál `tx_ready` bude mít hodnotu log. ,1' a může se přejít do dalšího stavu.
- **st21\_readaddinc** – čtecí adresa se inkrementuje, pokud ještě nebyly vyčteny všechny záznamy uložené v CRAM, přechází se do st11 a celý proces se opakuje. Pokud je vyčítání dokončeno, přechází se do st14.
- **st14\_send\_done** – signál `send_done` nastaven do log. ,1' a FSM přechází do výchozího stavu, což signalizuje MSM dokončení procesu vyčítání z paměti výsledků.

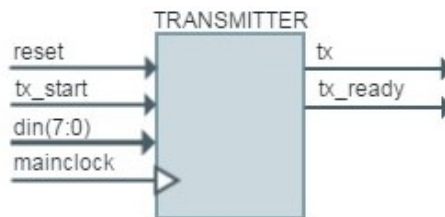


**Obrázek 39** Ukládání dat do CRAM



**Obrázek 40** Princip vyčítání a odesílá dat pře RS232

## 8.7 Transmitter



### 8.7.1. Vstupní porty:

- **Reset** – signál reset nuluje všechny čítače – baud\_rate\_counter, bit\_counter.
- **Tx\_start** – signál vystavuje komponenta „LOG“, pokud je v log. '1' povoluje zahájení přenosu dat pře sériovou linku.
- **Din** – data, která budou přes sériovou linku odesílána do počítače.

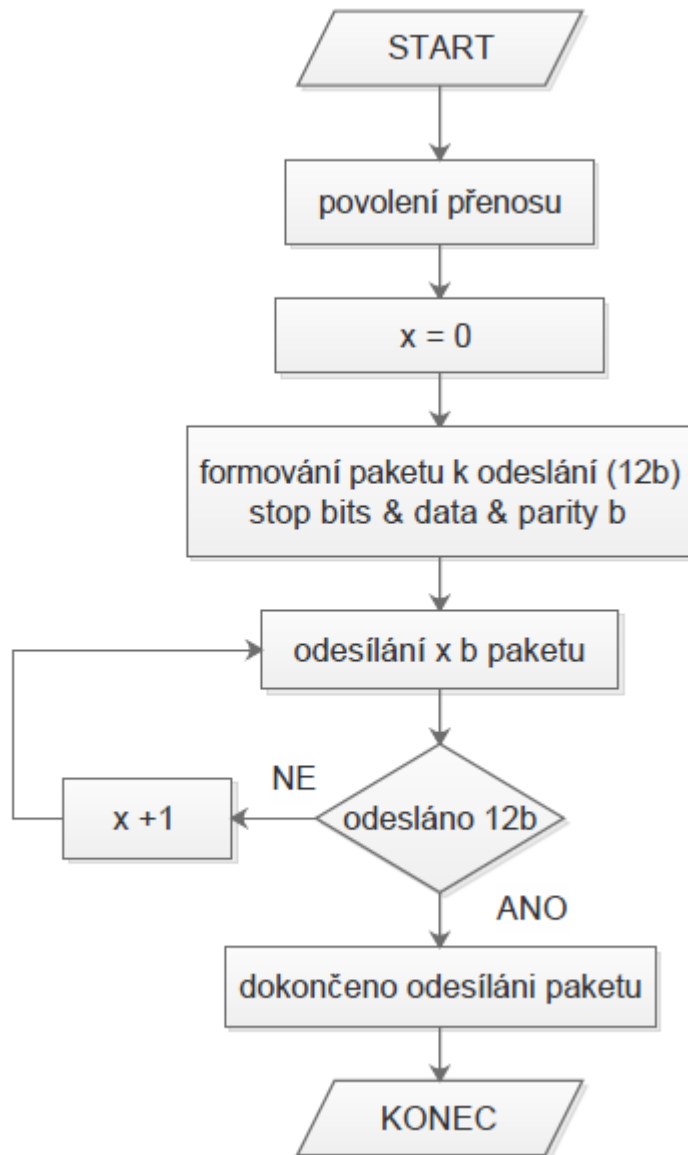
### 8.7.2. Výstupní porty:

- **Tx\_ready** – pokud je v log. '1' signalizuje dokončení odesílání dat.
- **Tx** – pin na desce NEXYS3, zde se posílají data, začíná se od nultého bitu.

### 8.7.3. Princip funkce:

Vysílač pracuje na **přenosové rychlosti** 1MBd. Data k odesílání jsou zformována komponentou „LOG“ do 8b signálu. V případě RS232 každé sekvenci datových bitů předchází jeden start bit, kterým se logická hodnota na lince přepne (původně v klidovém stavu) do opačného stavu. Po datových bitech následuje paritní bit a za ním jeden nebo více stop bitů, během kterých je linka opět v klidovém stavu. Data se tedy posílají po **12b paketech**. Odesílání bitů na pin tx se realizuje pomocí **posuvného registru**. Jakmile je odesílání 12b paketu dokončeno, je vystaven signál, který dá komponentě „LOG“ povolení k zformování dalšího 8b paketu.

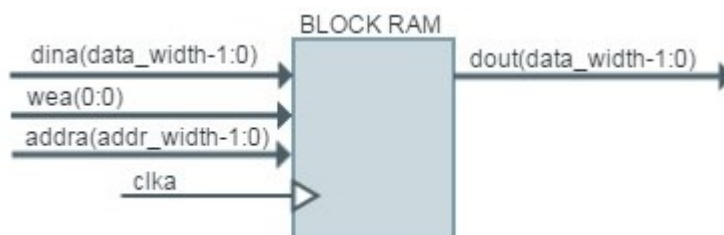




**Obrázek 41** Princip funkce vysílače

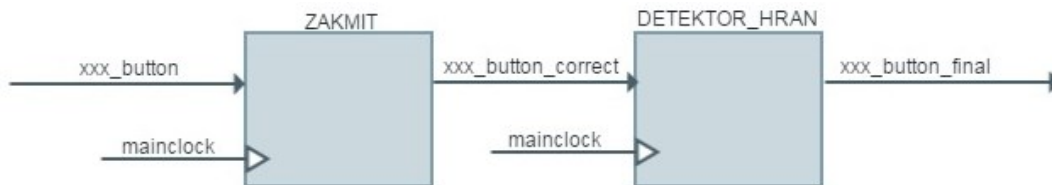
## 8.8 BRAM

Návrh obsahuje celkem tři blokové paměti, všechny slouží k ukládání aktuálního obsahu analyzovaných pamětí. Velikost BRAM odpovídá velikosti dané paměti.



**Obrázek 42** Entita BRAM

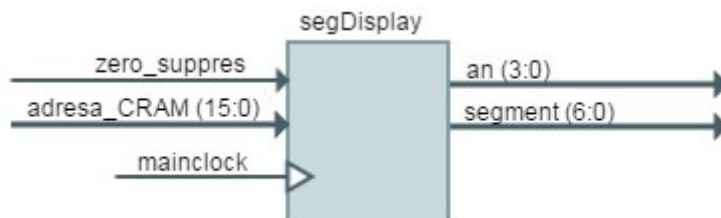
## 8.9 zakmit, detektor\_hran



Obrázek 43 Entity zakmit, detektor\_hran

Při spínání mechanických kontaktů často dochází ke krátkodobému odskočení kontaktů a vzniku několika impulzů, způsobujících nesprávné vyhodnocení jednotlivých sepnutí. Proto je nutné **zákmitům** předejít. V návrhu je použito programové ošetření zákmitu tlačítek, které vychází z časové prodlevy mezi vícenásobným čtením stavu spínače. Stav tlačítka bude akceptován, pokud se jeho stav nezmění ve dvou po sobě jdoucích čteních s časovou prodlevou. Jakmile je ošetřen zákmit na tlačítku, a správně detekována hrana signálu, je pomocí **detektor\_hran** vygenerován impuls o šířce jedné periody hodinového signálu.

## 8.10 segDisplay



Obrázek 44 Entita segDisplay

Řadič 7 segmentového displeje lze konfigurovat na různě široké displeje a je možné potlačit zobrazování úvodních nul. Desetinná tečka není využita. Na displeji se se bude zobrazovat aktuální **adresa paměti výsledků** – CRAM, tak bude možné již během probíhajícího měření odhadnout výsledek pokusu.

## 9. Simulace a implementace návrhu, testování přípravku

V této fázi projektu je hotov jak přípravek pro průběžnou analýzu obsahu pamětí, tak i návrh jeho obvodové řídicí logiky. Vzhledem k tomu, že je **návrh** obvodové logiky poměrně rozsáhlý a komplexní, není vyloučeno, že **může obsahovat chyby**. Implementace takového návrhu by mohla způsobit neočekávané, či nesprávné chování přípravku a v návaznosti na měření pak nesprávné ukládání dat případně i jejich ztrátu. V krajním může dojít k poškození přípravku nebo v důsledku pak i samotné desky FPGA. Je zřejmé, že je třeba takové situace předejít. Výhodou vývojového prostředí ISE Design Suite je, že zastaví implementaci návrhu, pokud jsou v syntaxi nějaké chyby a do upozornění vypíše místa návrhu, která by mohla představovat problém. Což ale zatím nezaručí, že i po odstranění chyb a ošetření upozornění, bude návrh fungovat přesně tak, jak návrhář zamýšlel. Proto je v prostředí ISE k dispozici simulátor **ISim**, který simuluje obvod z hlediska principu jeho činnosti, bez uvažování reálných vlastností obvodu. Vytvořením **testovacího souboru** (test bench) lze simulovat **chování** navrhovaného obvodu **za předem stanovených podmínek**. Pro zápis testovacího souboru se rovněž používá jazyk VHDL, stačí definovat hodnoty a časování vstupních a výstupních signálů.

### 9.1 Simulace programování SRAM

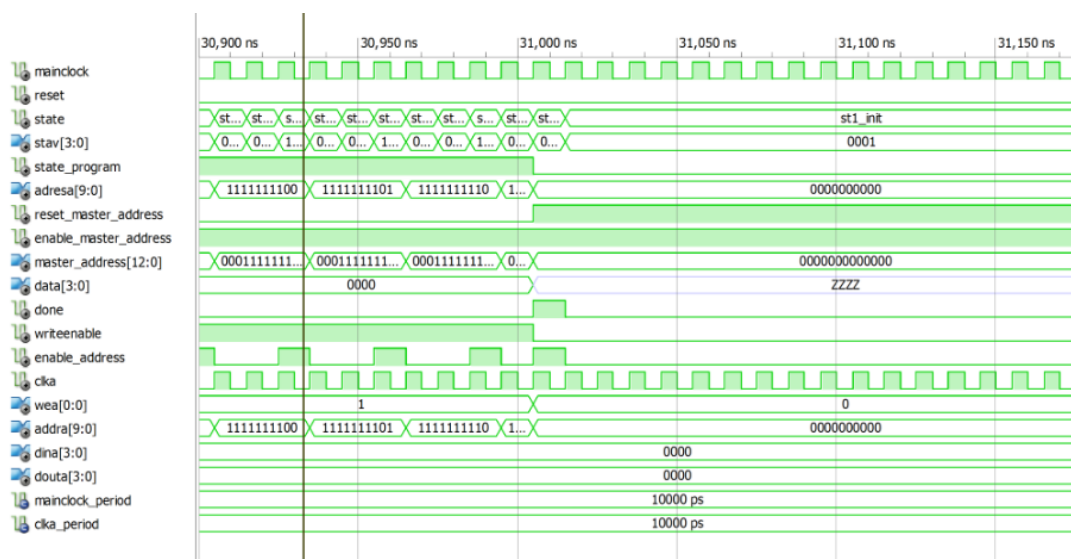
Simulace programování SRAM slouží především k otestování reakce stavového automatu „programování“ na stisk tlačítka reset a programování, a správnosti řízení hlavní adresy pro všechny paměti (master address). iSim umožňuje do simulace umístit libovolný počet značek (modrá barva) a ve sloupci „value“ vždy vypíše hodnoty všech signálů na aktuální pozici značky.



Obrázek 45 Simulace programování SRAM, záznam z iSim 0ns – 400ns

Z průběhu simulace je patrné, že když je signál **reset** v log. ,1‘ stavový automat je správně v počátečním stavu (st1\_init) a hodnoty všech signálů jsou v log. ,0‘. Automatu je vystaven signál povolení inkrementace hlavní adresy do log. ,1‘, ale reset hlavní adresy je zatím v log. ,1‘, čeká se na stisk tlačítka program.

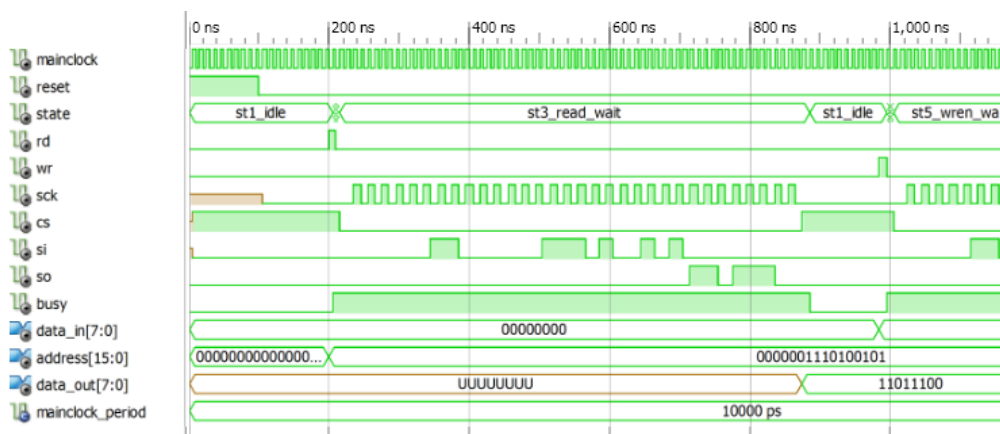
Po uvedení signálu **state\_program** do log. ,1‘ (stisk tlačítka program), reaguje FSM povolením programování (write\_enable log. ,1‘) a nastavením resetu hlavní adresy do log. ,0‘. Nyní bude hlavní adresa inkrementovat vždy po vystavení signálu enable\_address do log. ,1‘, který značí dokončení programování jedné buňky paměti.



Obrázek 46 Simulace programování SRAM, záznam z iSim 30900ns – 32000ns

Nyní se čeká na dokončení programování – vystavení **done** do log. ,1‘. Na simulaci lze ověřit, že po vystavení done do log. ,1‘ a nastavení state\_program do log. ,0‘, reaguje FSM nastavením zrušením povolení zápisu a resetem hlavní adresy, tak aby se mohlo buď znovu programovat, nebo číst od nulté adresy.

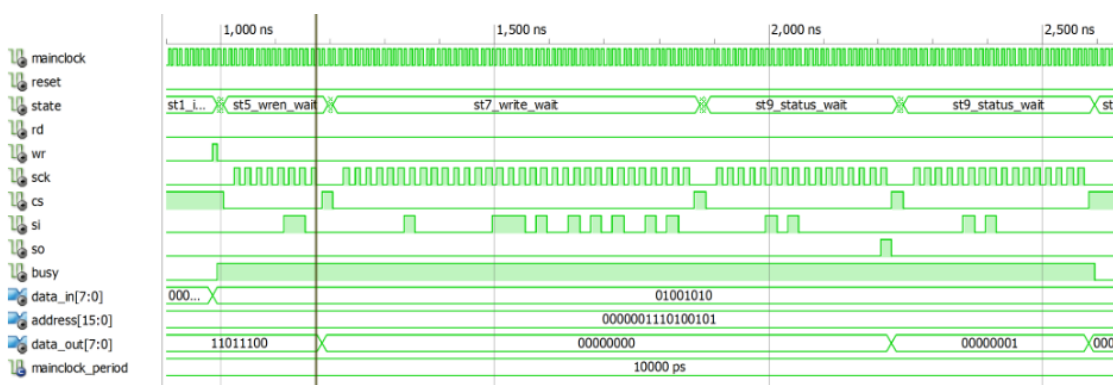
## 9.2 Simulace programování a čtení EEPROM



Obrázek 47 Simulace čtení EEPROM, záznam z iSim 0ns -1500ns

Simulace programování nebo čtení z paměti se sériovým rozhraním je v porovnání s paralelním rozhraním komplikovanější z toho důvodu, že v případě sériového rozhraní je někdy nutné počkat na odpověď paměti a teprve poté se může provádět další instrukce. Proto musí testovacího souboru obsahovat i simulace odpovědi paměti.

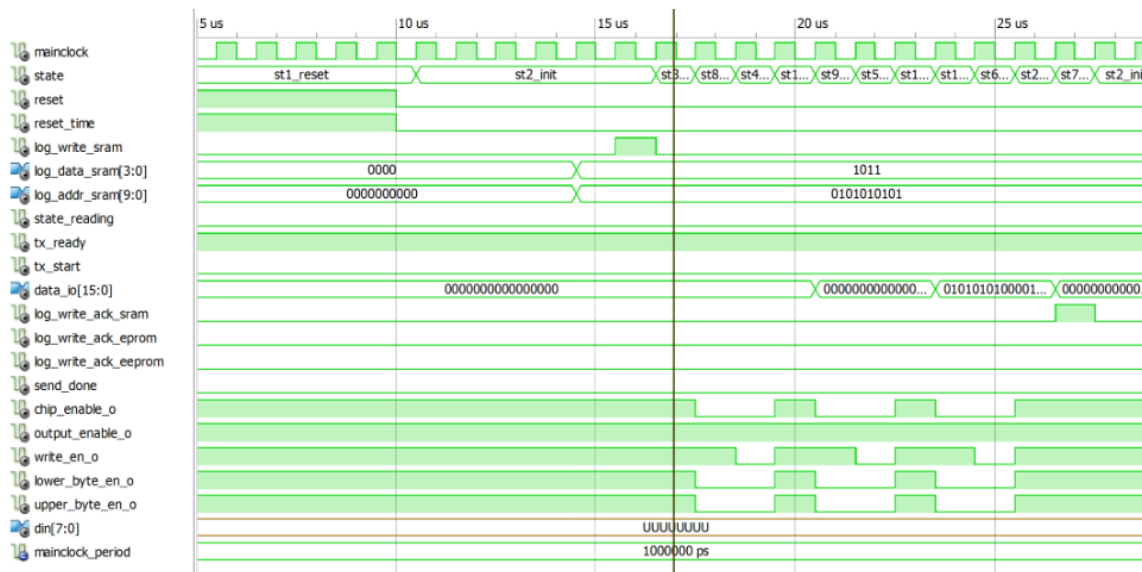
Nejprve bylo testováno čtení (**Obrázek 47**) a to nastavením adresy na hodnotu X"03A5" a uvedením signálu **rd do log. '1'**. Na simulaci lze ověřit, že FSM správně reaguje: paměť je zaneprázdněna - busy v log. '1', CS do log. '0' – započetí komunikace s pamětí, spuštění sériových hodin a následuje posouvání paketu dat na pin SI (instrukce READ, adresa). Poté je na pinu SO nasimulována odpověď a FSM přivádí CS do log. '1' (zrušení komunikace s pamětí) a zastaví sériové hodiny.



**Obrázek 48** Simulace programování EEPROM, záznam z iSim 900ns – 2600ns

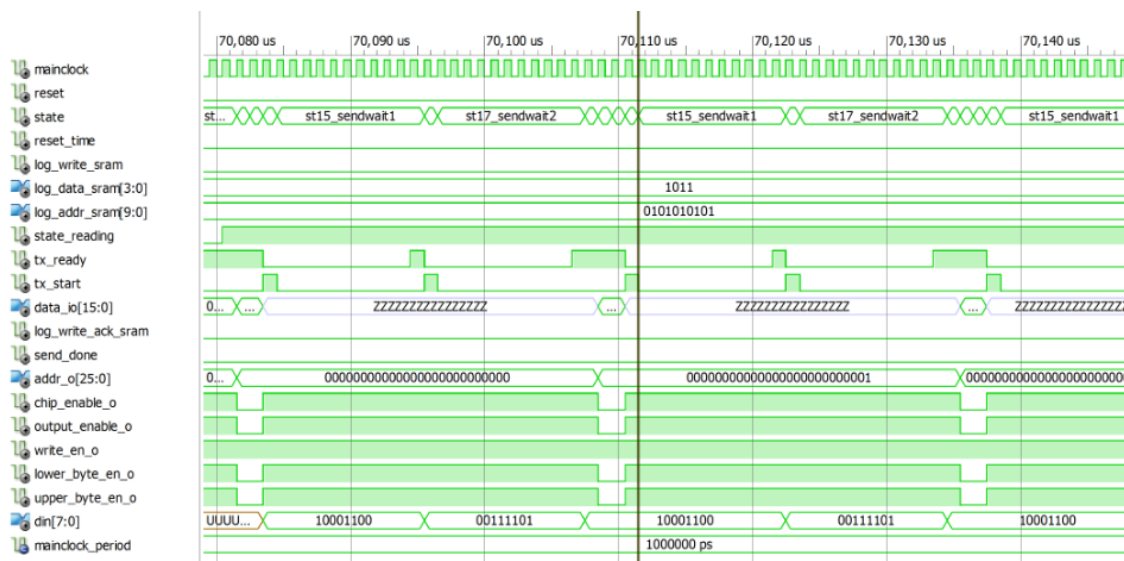
Poté se testuje funkčnost **zápisu** dat X"4A" na adresu X"03A5" uvedením signálu **wr do log. '1'**. Následuje žádoucí reakce FSM: paměť je zaneprázdněna - busy v log. '1', CS do log. '0' (započetí komunikace s pamětí), spuštění sériových hodin. Nejprve je odeslána instrukce povolení zápisu (WREN), nečeká se na odpověď a CS nastaveno zpět do log. '1', jsou zastaveny sériové hodiny. CS jde do log. '0', spustí se sériové hodiny, na SI je posouván datový paket (instrukce WRITE, adresa, data), po odeslání všech bitů se nečeká na odpověď paměti. Proces dále pokračuje čtením stavového registru, pro ověření dokončení zápisu. Na SI pin je odeslána instrukce čtení stavového registru. Na pinu SO je nasimulována odpověď – zápis probíhá. FSM odesílá instrukci znovu, tentokrát je simulována odpověď- zápis dokončen. CS přechází do log. '1', sériové hodiny jsou zastaveny, paměť již není zaneprázdněna – busy v log. '0'.

### 9.3 Simulace funkce „LOG“



**Obrázek 49** Simulace ukládání záznamů, záznam z iSim 5 $\mu$ s – 30 $\mu$ s

Simulace ukládání záznamu je realizována pro všechny tři paměti, zde se uvádí pouze signály paměti SRAM pro přehlednost. V simulaci se opět testuje reakce FSM „LOG“ na signál reset. Pro každou paměť byla libovolně zvolena adresa buňky a data, představující detekovanou chybu. Nastavením **log\_write\_sram do log. ,1‘** se simuluje zjištění chyby. Z **Obrázek 50** je zřejmé, že po zjištění chyby FSM řídí signály paměti CRAM a probíhá ukládání paketů dat do paměti. Jakmile je ukládání dokončeno, signál **log\_write\_ack\_sram** je vystaven do **log. ,1‘** a FSM se vrací do výchozího stavu a čeká na další detekci chyby. Je vidět, že celý proces probíhá velmi rychle, ukládání jednoho záznamu trvá asi 30us.



**Obrázek 50** Simulace čtení dat z paměti výsledků CRAM a odesílání dat do počítače, záznam z iSim  
70080 $\mu$ s - 70150 $\mu$ s

Proces odesílání dat přes sériovou linku do počítače zahrnuje také čtení dat uložených v paměti výsledků. Proces odesílání se aktivuje nastavením **state\_reading do log. ,1‘**, FSM nastaví odpovídající signály CRAM, která vystaví uložená data na data\_io. Čeká se, dokud není komponenta „*transmitter*“ připravena odesílat data – simuluje se nastavením signálu tx\_ready do log. ,1‘, pak FSM vysílá příkaz k započetí odesílání - tx\_start nastaví do log. ,1‘. Během odesílání je data\_io ve stavu vysoké impedance. Jakmile jsou odeslány pakety dat, adresa CRAM je inkrementována o 1 a proces probíhá znovu. Pokud jsou odeslána všechna data uložená v paměti výsledků, signál send\_done je vystaven do log. ,1‘.

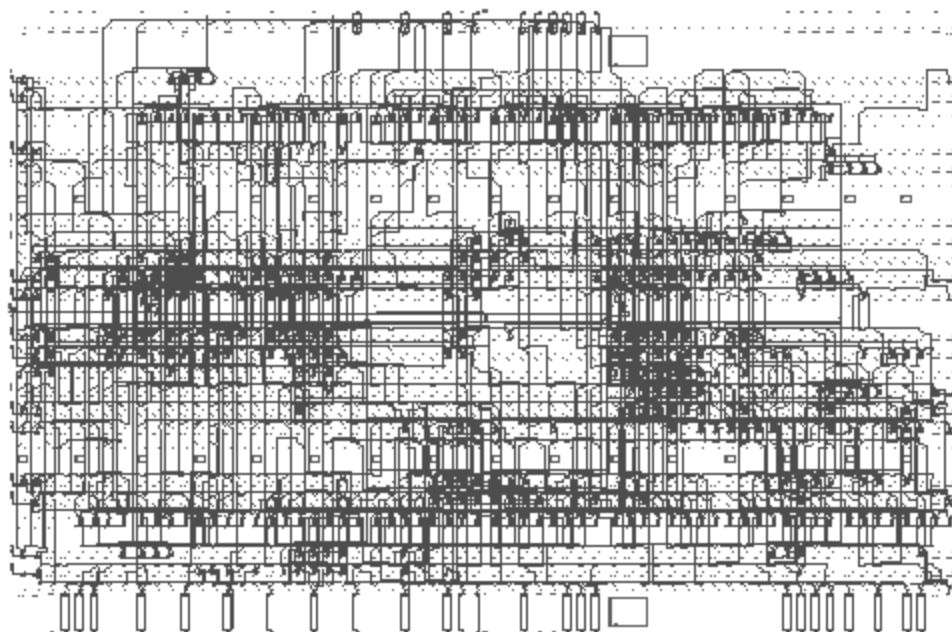
Tímto je ověřena správná funkčnost nejdůležitějších modulů návrhu teoreticky a nyní lze provést jeho implementaci.

## 9.4 Implementace návrhu do FPGA

Ke konfiguraci cílového zařízení, v tomto případě Spartan6 XC6LX16-CS324, je nejprve nutné provést **syntézu** zdrojového souboru zapsaného v jazyce VHDL. Výsledkem syntézy je tzv. **netlist**, který lze popsat jako obecný seznam elementárních sekvenčních a kombinačních logických obvodů a jejich propojení, které budou realizovat navrženou funkci. Jakmile je syntéza dokončena, návrhové prostředí zahájí proces **implementace**, který je již závislý na cílové součástce. Implementace má tři fáze: mapování, rozmístění a propojení. **Mapování** nahradí obecné logické prvky těmi, které jsou součástí konkrétní architektury FPGA (kombinační logika se namapuje do LUT, sekvenční do DKO). Fáze **rozmístění a propojení** prvků do sktruktury čipu je ze všech stadií implementace časově nejnáročnější, protože se musí najít vhodné propojení mezi prvky. Po dokončení implementace je vygenerován nový netlist, který už obsahuje zcela konkrétní prvky a propojení cílové platformy. Nyní se z netlistu vytvoří **binární konfigurační soubor**, který je pomocí USB kabelu a Digilet Adept (aplikace, která umožňuje konfigurovat logická zařízení firmy Xilinx) nakonfigurován do čipu FPGA. V tomto případě trval celý proces až po generování binárního souboru asi 110s. [25]

Využití zařízení			
Využití logických vrstev	Použité	Dostupné	Využití
Klopné obvody	641	18,224	3%
LUT	757	9,112	8%
RAMB16B	18	32	56%
RAMB8B	1	64	1%
DCM/DCM_CLKGENs	1	4	25%

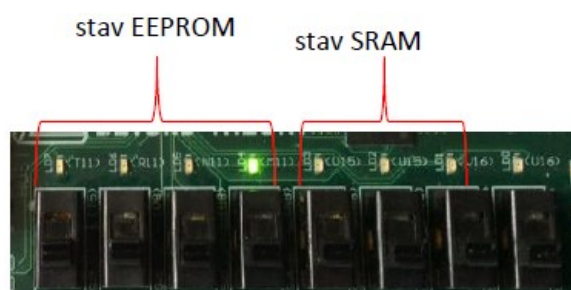
**Tabulka12** Využití zařízení [%], tabulka upravena podle návrhového prostředí ISE Design Suite.



Obrázek 51 Implementovaný návrh z vnitřního pohledu FPGA

## 9.5 Testování přípravku

Po naprogramování desky FPGA je nyní nutné ověřit správnou funkčnost návrhu i s reálnými obvody. Jak již bylo zmíněno v předchozí kapitole, součástí návrhu jsou prostředky **zpětné vazby** probíhajících procesu - **led** indikující aktuální stav FSM „programování“, který se může nacházet v 11 stavech, takže ke kódování aktuálního stavu bude stačit 4b číslo. K dispozici je sedm diod: **led (7:4)** zobrazují stav programování EEPROM a **led (3:1)** náleží SRAM.



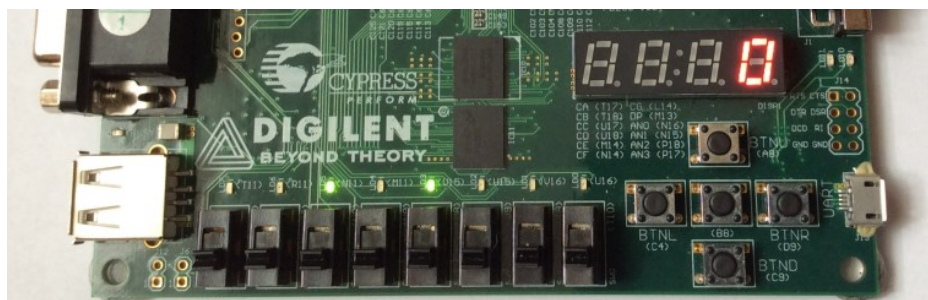
Obrázek 52 Led na desce FPGA používané k indikaci aktuálního stavu programování

Obvodová logika přípravku pracuje s frekvencí 1MHz, FSM některými stavy prochází, tak rychle, že není možné pouhým okem zaznamenat přeblikávání led. Pro ověření úspěšného naprogramování paměti však postačí, když je vidět, že se FSM programování obou pamětí vrátí do výchozího stavu (init, b0010).

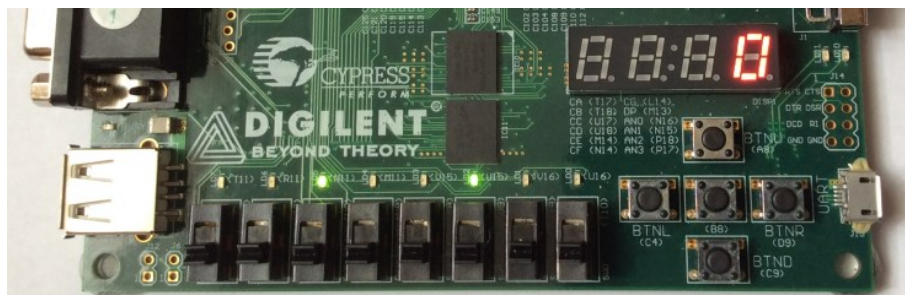
Při programování **SRAM** s paralelním rozhraním se na žádnou odpověď paměti nečeká (např. nemusí se číst stavový registr), paměť má malou velikost, takže **programování** probíhá velmi **rychle** – paměť o velikosti 4Kb se programuje 3,31s.



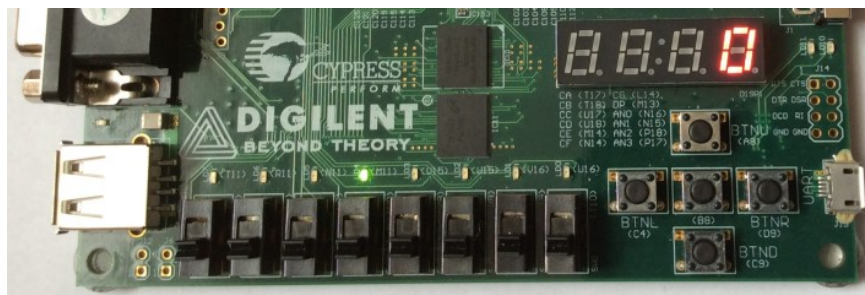
U **EEPROM** se sériovým rozhraním bude probíhat programování podstatně **pomaleji**, konkrétně paměť o velikosti 64Kb se programuje 24,44s.



**Obrázek 53** Stav těsně po stisku tlačítka program. SRAM (stav b“1000“ incr1), EEPROM (stav b“0010“ next\_addr\_program) – led indikují velmi rychlou inkrementaci hlavní adresy, proto se zdá, že dioda 5 a 3 neustále svítí.



**Obrázek 54** Programování SRAM je dokončeno (stav b“0100“ done), programování EEPROM stále probíhá.

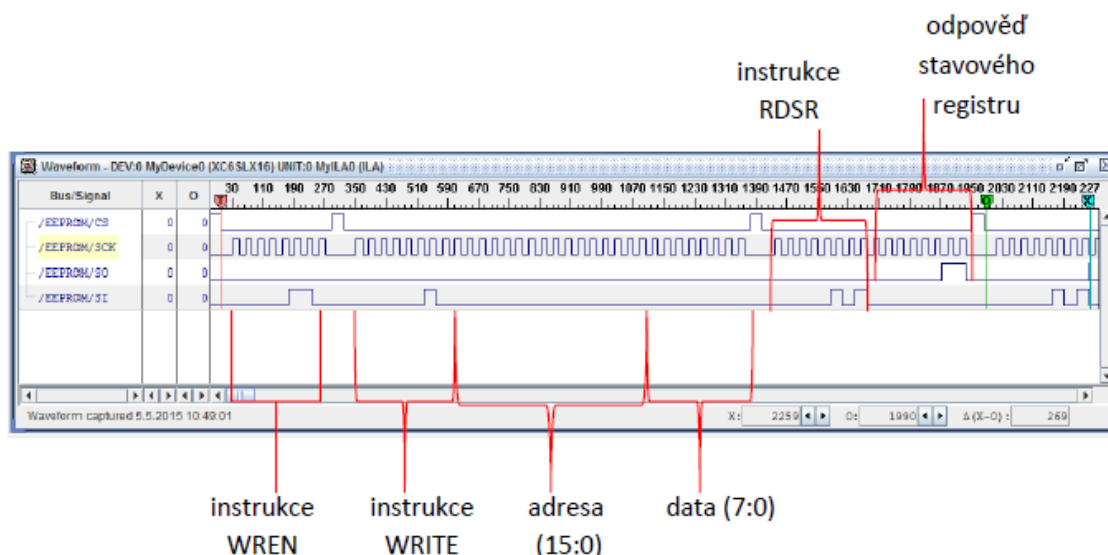


**Obrázek 55** Programování obou pamětí je dokončeno, oba FSM „programování“ se nacházejí ve výchozím stavu (stav b“0001“ init).

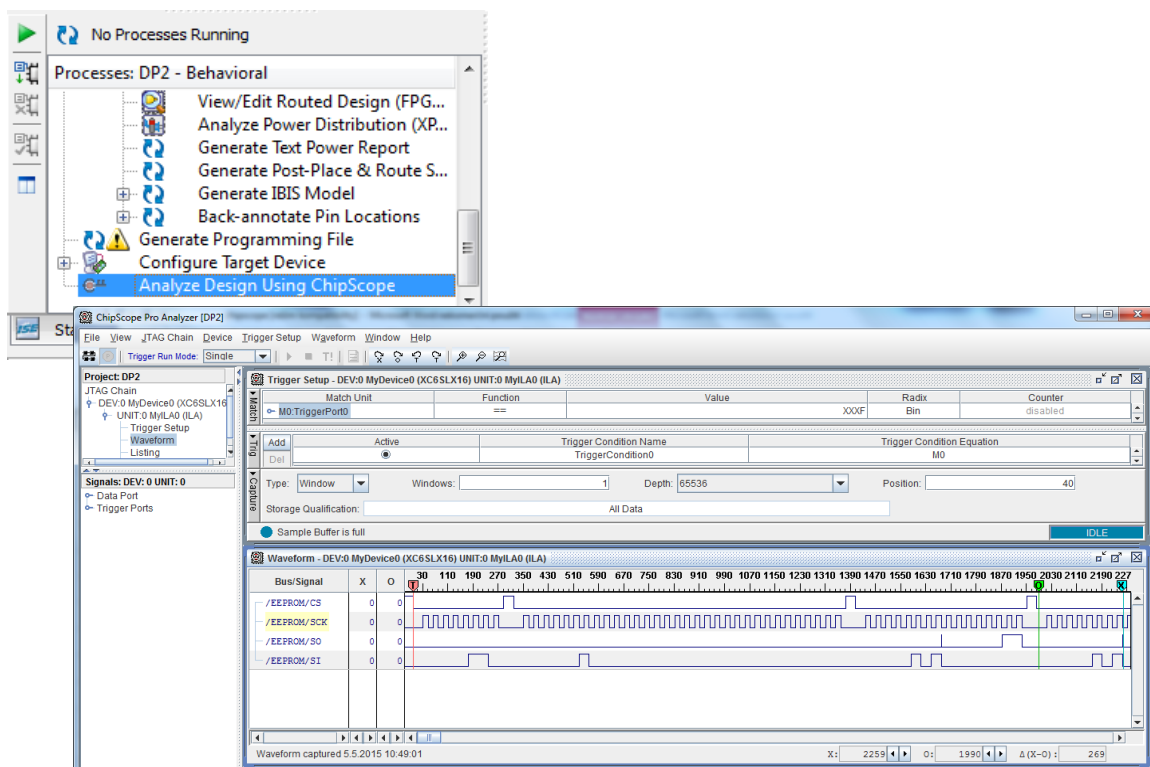
Návrhové prostředí ISE Design Suite umožňuje do návrhu obvodové logiky vložit také **logický analyzátor** a jeho prostřednictvím pak sledovat reálný průběh libovolného signálu uvnitř návrhu. Signály jsou zachyceny uvnitř systému a následně vyvedeny přes programovací rozhraní, takže není potřeba další sondy. Zachycené signály jsou pak analyzovány nástrojem **ChipScope Pro Analyzer**. (viz **Obrázek 57**)

Použití logického analyzátoru je vhodné pro **zobrazení procesu komunikace s EEPROM**, který je poměrně komplikovaný a jeho správnost je ověřena zatím pouze teoreticky pomocí simulace.

Z **Obrázek 56** je patrné, že průběh reálných signálů odpovídá průběhům signálů namodelovaných v simulaci, deska FPGA s přípravkem komunikuje správně. Přípravek a je tedy připraven k měření.

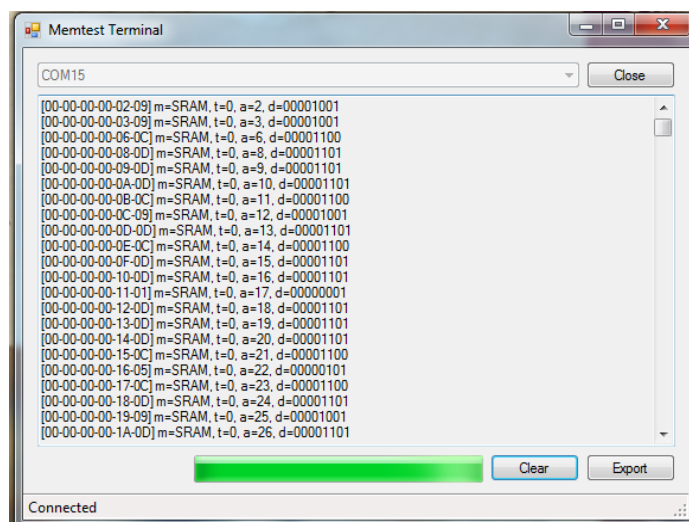


**Obrázek 56** Ověření funkce programování EEPROM, záznam z ChipScope Pro



**Obrázek 57** Spuštění ChipScope Pro z vývojového prostředí ISE, dialogové okno Chipscope Pro

## 10. Vizualizace výsledků



Obrázek 58 Dialogové okno MemtestTerminal

### 10.1 MemtestTerminal

Terminal pro příjem dat do počítače z desky NEXYS3 přes sériovou linku byl vytvořen v programu Microsoft Visual Studio Express 2012 v programovacím jazyce C#.

Pokud je k počítači připojeno zařízení přes **sériový port** (COM15), terminál je schopen daný port detekovat a po kliknutí na tlačítko „*Open*“ se k portu **připojit**, tlačítko se změní na „*Close*“. Zda připojení proběhlo úspěšně lze ověřit v levém dolním rohu, kde je zobrazen stav připojení „*Not Connected*“, nebo „*Connected*“.

Nyní může uživatel odeslat stisknutím tlačítka odesílání na desce NEXYS3 data terminálu. Probíhající **příjem dat** Terminálem signalizuje hlášení „*Process running...*“ vlevo dole. Jakmile hlášení zhasne, začíná terminál proces **formování dat**, uložených v CRAM v binárních hodnotách, na hodnotitelná data:

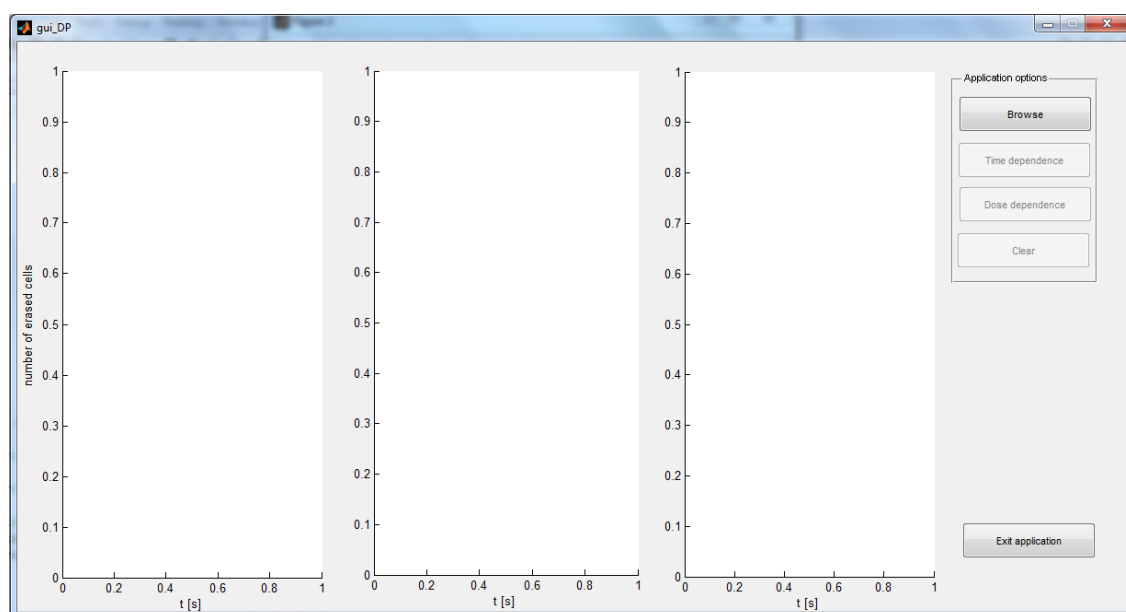
- Binárnímu kódu paměti je přiřazen skutečný typ paměti: SRAM, EPROM, EEPROM.
- Čas záznamu je zobrazen jako desetinné číslo v sekundách s přesností na setiny sekundy.
- Adresa buňky a změněná data zůstávají v binárním formátu.

**Doba formování** dat je přímo úměrná objemu dat uložených v CRAM, stav procesu je zobrazen na postupové liště dole uprostřed.

Aby bylo možné data dále analyzovat, lze přijatá a zformovaná data překopírovat do \*.csv souboru, a to stiskem tlačítka „Export“. Otevře se dialogové okno, uživatel zvolí místo uložení souboru, soubor pojmenuje a musí k názvu připojit „.csv“.

Pokud chce uživatel provést nové měření a poslat Terminálu jiná data, použitím tlačítka „Clear“ může obsah terminálu **vymazat**. Ale pokud již žádná další měření nebudou realizována, kliknutím na tlačítko „Close“ může port **zavřít**.

## 10.2 GUI\_DP



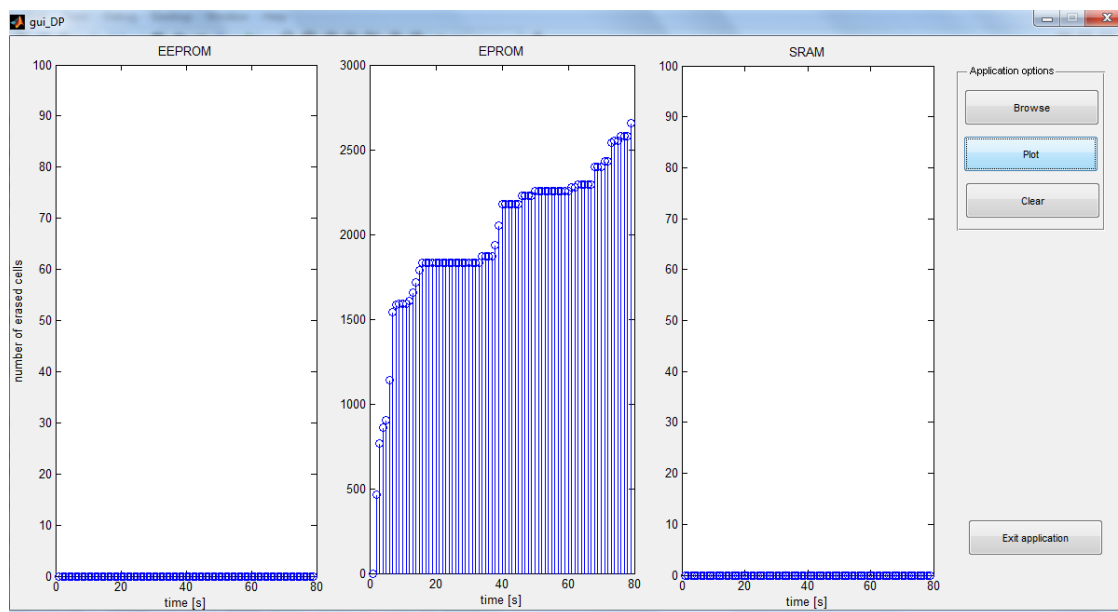
Obrázek 59 Dialogové okno GUI\_DP

Nástroj GUI\_DP pro **vizualizaci výsledků** je navržen v programu Matlab. Tento jednoduchý program umožňuje pomocí tlačítka „Browse“ vyhledat a načíst Terminálem vytvořený \*.csv soubor. Jakmile jsou data načtena, tlačítkem „Time dependence“ nebo „Dose dependence“ se do připravených souřadných systémů **zobrazí počet smazaných buněk v závislosti na čase, nebo na dávce**. Tlačítkem „Clear“ lze grafy smazat. Pokud uživatel nepotřebuje s programem dále pracovat, stisknutím tlačítka „Exit application“ program zavře.

## 11. Kontrolní měření

Pro ověření funkčnosti celého projektu, bylo provedeno kontrolní měření. Ze tří analyzovaných pamětí je EPROM jediná, na které se dá simulovat proces mazání v čase, a to použitím UV záření.

Paměť EPROM byla po dobu přibližně 1,5 min vystavena UV záření z mazačky EPROM, během této doby byly zaznamenávány změny v obsahu paměti. **Obrázek 61** ilustruje výsledek kontrolního měření.



**Obrázek 60** Výsledek působení UV záření na EPROM

	A	B	C	D	E	F	G
1	raw	memory	timestam	address	data		
2	20-00-00-(EPROM	0.5		0	11110101		
3	20-00-00-(EPROM	0.5		16	10000000		
4	20-00-00-(EPROM	0.5		8	11110101		
5	20-00-00-(EPROM	0.5		24	11110101		
6	20-00-00-(EPROM	0.5		4	11000000		
7	20-00-00-(EPROM	0.5		20	11110101		
8	20-00-00-(EPROM	0.5		12	11110101		
9	20-00-00-(EPROM	0.5		28	11000000		
10	20-00-00-(EPROM	0.5		32	11110101		
11	20-00-00-(EPROM	0.5		48	11110101		
12	20-00-00-(EPROM	0.5		40	11110101		
13	20-00-00-(EPROM	0.5		56	11000000		
14	20-00-00-(EPROM	0.5		36	11110101		
15	20-00-00-(EPROM	0.5		52	11110101		
16	20-00-00-(EPROM	0.5		44	11110101		
17	20-00-00-(EPROM	0.5		60	11000000		
18	20-00-00-(EPROM	0.5		64	11110101		
19	20-00-00-(EPROM	0.5		80	11110000		
20	20-00-00-(EPROM	0.5		72	11000000		
21	20-00-00-(EPROM	0.5		88	11110101		
22	20-00-00-(EPROM	0.5		68	11110101		
23	20-00-00-(EPROM	0.5		84	11110101		
24	20-00-00-(EPROM	0.5		76	11110101		
25	20-00-00-(EPROM	0.5		92	11110101		

**Obrázek 61** Část zaznamenaných dat

## 12. Experimentální měření

Experimentální měření bylo realizováno na dvou přístrojích: TERaGAM Cobalt Unit a CyberKnife. Před zahájením měření bylo třeba vyřešit rozsah celkové absorbované dávky<sup>10</sup> dodané přípravku během experimentu. Tato dávka by měla být v souladu s rozsahem hodnot běžně používaných v klinické praxi.

„Cílem radioterapie je selektivní **likvidace nádorového ložiska** při co nejmenším poškození okolních zdravých tkání. Ozáření okolních tkání se přitom nikdy nedá zcela vyhnout, je však třeba dodržet tzv. **toleranční dávku** v kritických tkáních a orgánech. Do cílové oblasti je třeba zavést dostatečně **vysokou dávku záření**, pro nádorové buňky **letální** - tumorózní kanceroletální dávku (cca **50-150Gy**) takovým způsobem, aby okolní zdravé tkáně nebyly enormně poškozeny. Úkolem radioterapie v klinické praxi je najít **optimální kompromis** mezi těmito dvěma protichůdnými požadavky.“ [27]

### 12.1 Ozařovač TERAGAM

Ozařovače TERAGAM se používají k léčbě zhoubných nádorů. Využívají ionizujícího záření uvolňovaného ze zdroje představovaného zářičem Co-60. Výhodou těchto ozařovačů je možnost ozařování tumoru nejen staticky, ale i pohybově (rotačně). Kobaltový zdroj generuje záření  $\gamma$  o energiích přibližně 1,17 a 1,33 MeV. [25] Doba, za kterou TERAGAM dodá hmotě dávku 1Gy, je 1min a 20s.



**Obrázek 62** Kobaltový ozařovač (vlevo), detail kolimátoru v ozařovací hlavici (vpravo), převzato [26]

<sup>10</sup> „**Absorbovaná dávka** ionizujícího záření je fyzikální veličina, která udává energii dodanou jednotkovému množství hmoty průchodem příslušného záření. Jednotkou absorbované dávky záření je Gray (Gy), rozměrově se jedná o joule na kilogram.“ [22]



## 12.2 CyberKnife

Jedná se o stereotaktické radiochirurgické zařízení, které léčí nádory v různých oblastech těla. Zdrojem  $\gamma$  záření je kompaktní **lineární urychlovač** elektronů o energii až 6MeV s terčíkem, který konvertuje energii elektronů na brzdné  $\gamma$ -záření. **Úzké kolimátory** umožňují nastavení různých průměrů ozařovacího svazku. „Používá se buď sada mechanicky **výměnných kolimátorů** s různě velkými otvory, nebo je kolimátor opatřen proměnnou **irisovou clonou**, jejíž elektricky pohyblivé segmenty umožňují automatické on-line nastavení různých apertur - průměrů ozařovacího svazku během ozařování“ [27]. CyberKnife ozařuje s 10x-30x větší přesností než jiné lineární urychlovače. Pomocí robotického ramene je schopen vysílat svazek velmi tenkých paprsků z téměř každé pozice. Ozařované místo i robotické rameno jsou nepřetržitě monitorovány a kontrolovány počítačem. „U **stereotaktické radioterapie** se do malého cílového ložiska aplikuje vysoká jednotlivá dávka (řádově desítky Gy) buď jednorázově, nebo v několika málo frakcích (1-5 frakcí).“ [27] Dávkový příkon je 800MU/min<sup>11</sup> což znamená, že za referenčních podmínek se naměří v maximu 8Gy za 1minutu záření.



Obrázek 63 CyberKnife, upraveno podle [28]

<sup>11</sup> MU (Monitor Unit) – je mírou ionizace, přítomné v léčebném paprsku v ozařovací hlavici. Jedna monitorovací jednotka je typicky rovna specifické dávce radiace, ve specifické hloubce vodního fantomu, pro paprsek s určitou energií, v určité velikosti pole a v určité vzdálenosti od cíle. Monitorovací jednotky se používají k měření výstupu přístroje, aby byla doručena do cílové oblasti správná dávka. 100MU odpovídá 1Gy absorbované dávky. [29]

## 12.3 Realizace měření

Měření bylo realizováno v ozařovacích onkologické kliniky Fakultní nemocnice Ostrava.

Před prvním měřením byla **hypotéza** následující: pokud je možné paměť EPROM smazat pomocí UV záření, pak rentgenové záření s mnohem kratší vlnovou délkou, bude mít na paměť obdobný účinek. Výsledek pokusu pro zbývající paměťové součástky nebylo možné před realizací pokusu nijak odhadnout. Jak je patrné z **Obrázek 61**, první změny v paměti EPROM při ozařování UV byly detekovány již během prvních sekund. První zkušební měření proto bylo realizováno s nízkou dávkou **2Gy**. Nejprve se pracovalo s ozařovačem **TERAGAM**.

### 12.3.1. TERAGAM



**Obrázek 64** Přípravek pod ozařovačem TERAGAM



**Obrázek 65** Detail přípravku, ozařovaná oblast je označena červeně



Přípravek byl umístěn přibližně 30cm pod ozařovací hlavici a pomocí kolimátorů bylo nastaveno ozařované pole tak, aby pokrylo všechny tři paměti (viz **Obrázek 65**). Obsah všech pamětí se naprogramoval na hodnoty log. „0“, na desce FPGA byl spuštěn proces kontroly a všichni účastníci měření opustili místnost. Přístroj TERAGAM se spouští z ovladovny, aby obsluha neobdržela zbytečně dávku záření. Jelikož se kobaltový ozařovač používá pro méně náročné ozařovací techniky, je ozařovací místnost vybavena pouze základní monitorovací technikou bez možnosti detailně sledovat ozařovací stůl.

Během probíhajícího pokusu tedy nebylo možno sledovat, zda v pamětech probíhají nějaké změny. Při dávce **2Gy** trvalo ozařování **2 min 40s**. Dávka 2Gy nezpůsobila v pamětech **žádnou změnu**.

Při druhém měření se dávka zvedla na **8Gy**, ozařování tedy trvalo **10 min a 40s** a ani během této doby či dávky nebyla detekována **žádná změna**. Poslední měření na přístroji TERAGAM bylo realizováno s dávkou **15Gy**, záření bylo aplikováno **20min**. Avšak ani v tomto případě nebyly zaznamenány **žádné změny**. Pro ověření byl reálný obsah paměti EPROM zkontrolován také pomocí programátoru, který je schopen zobrazit její aktuální obsah.

Na základě **nulových výsledků** realizovaných pokusů s přístrojem TERAGAM, byl stanoven **další cíl** – zjištění **hraniční hodnoty absorbované dávky**, při které již bude možné detekovat jakékoli změny. Z důvodů časové náročnosti a nemožnosti sledovat průběh experimentu s přístrojem TERAGAM, byla další měření realizována na přístroji CyberKnife, který je schopen dodat hmotě stejnou dávku záření téměř **8x** rychleji. Ozařovna je navíc vybavena **kvalitní monitorovací technikou** – čtyři kamery mohou detailně sledovat libovolné místo v místnosti.

### 12.3.2. CyberKnife

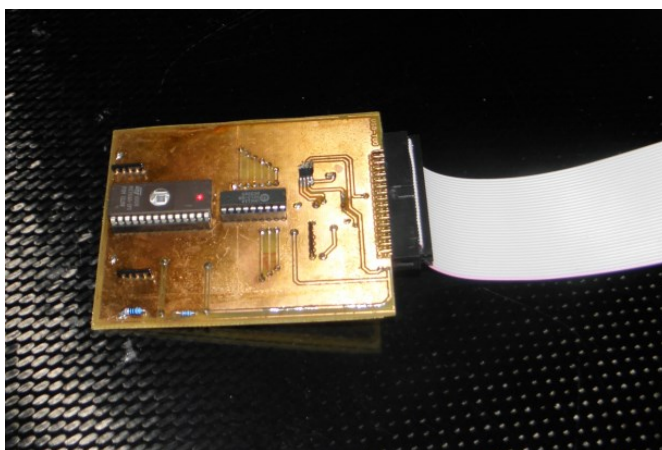
Aby byly ozářeny všechny paměti umístěné na přípravku, je třeba, aby měl svazek záření průměr alespoň 6cm. Hlavice CyberKnife je opatřena kolimátorem s irisovou clonou. Limitní hodnota při úplném uzavření clony je 0,25mm a při úplném otevření je 68mm. Což bylo pro tento pokus postačující. Přípravek byl umístěn pod ozařovací hlavici ve vzdálenosti přibližně 30cm, nyní se pod něj navíc umístil film (**Obrázek 69**), který po ukončení měření sloužil k ověření velikosti ozařovaného pole. Po naprogramování všech pamětí a spuštění procesu kontroly, všichni účastníci pokusu opustili ozařovací místnost. V ovladovně obsluha nastavila první kameru tak, aby bylo možné detailně sledovat displej FPGA desky s načítajícími se záznamy. Druhá kamera byla zaostřena na samotný přípravek, pokud by došlo při vysokých dávkách k přehřívání přípravku, pokus by se mohl včas ukončit.



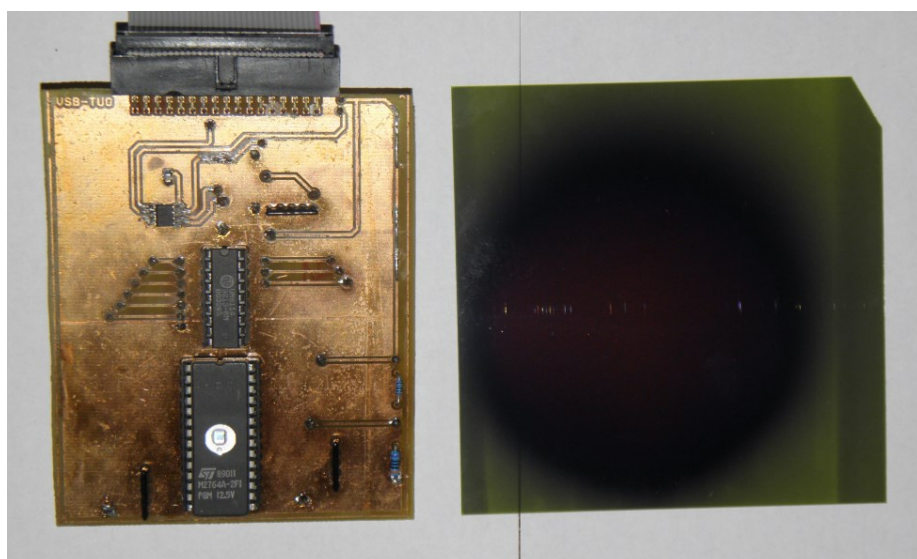
**Obrázek 69** Přípravek pod hlavicí CyberKnife



**Obrázek 68** Hlavice CyberKnife s irisovou clonou



**Obrázek 67** Detail přípravku, červený bod označuje střed ozařovaného pole.



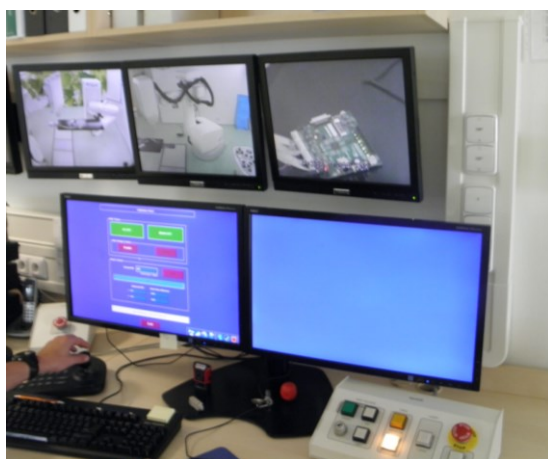
**Obrázek 66** Detekce velikosti ozařovaného pole na filmu.

Počáteční testovací hodnota na přístroji CyberKnife byla **20Gy**, záření trvalo **2min 30s**. Jelikož se ani v tomto případě neprojevil vliv záření na obsah paměti, dávka se zvedla na hraniční hodnotu **2000Gy** a na monitoru kamery se sledovalo, zda probíhají změny. Experiment bylo tedy možné podle potřeby ukončit.

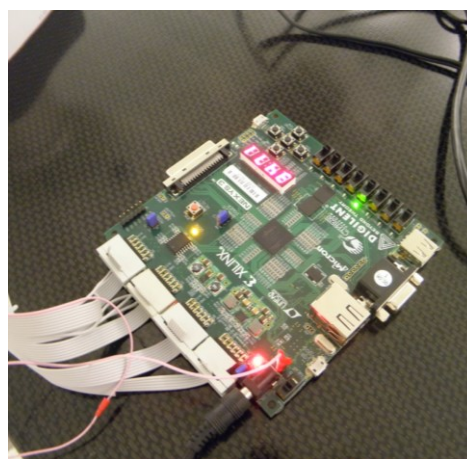
## 12.3.3. Výsledky

K první změně došlo až při dávce přibližně **165Gy** v čase **20min** od počátku ozařování a to v paměti EPROM. Frekvence výskytu změn v pamětech nebyla konstantní, od hodnoty **198Gy** začalo množství změněných buněk růst rychleji. Měření bylo ukončeno, jakmile se dosáhlo absorbované dávky **220Gy**.

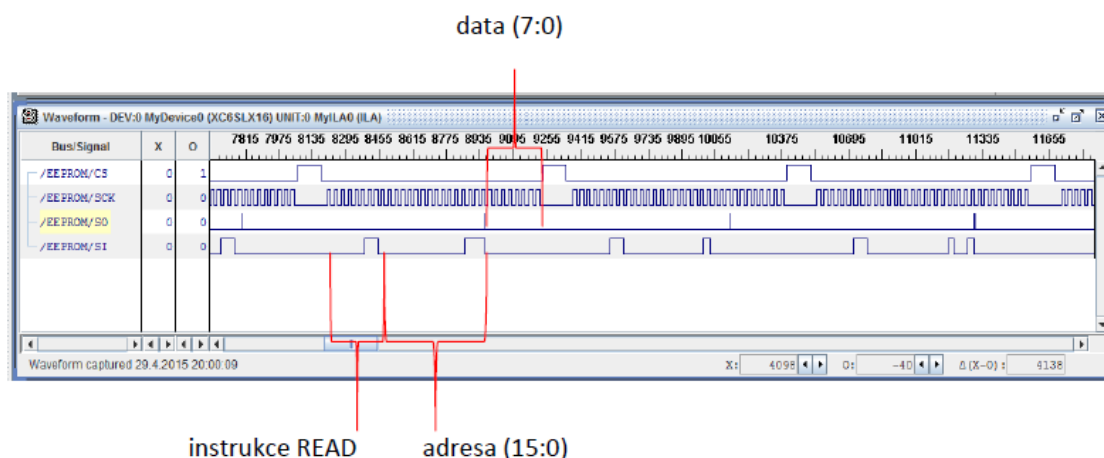
Během měření bylo zaznamenáno celkem **1660 změn** v paměti **EPROM**. Paměti **SRAM** a **EEPROM** nebyly během experimentu ionizujícím zářením **nijak ovlivněny**. Což bylo ověřeno přečtením obsahu paměti EEPROM pomocí ChipScope (logického analyzátoru) vývojového prostředí ISE.



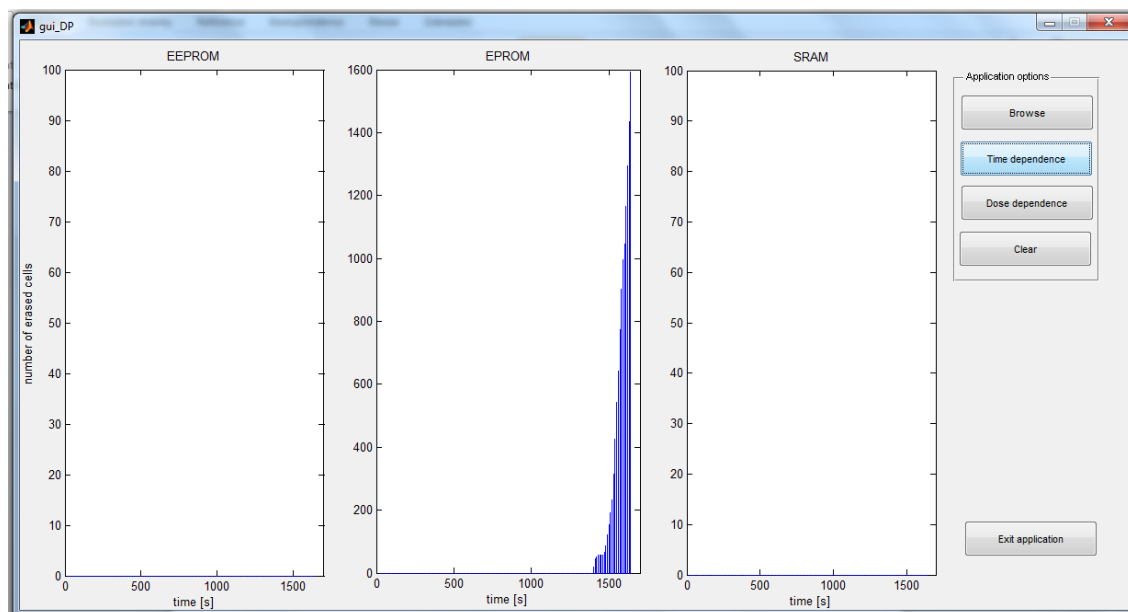
**Obrázek 71** Ovladovna CyberKnife, horní monitory jsou výstupy z kamer, na spodním monitoru je zobrazen stav aplikace záření.



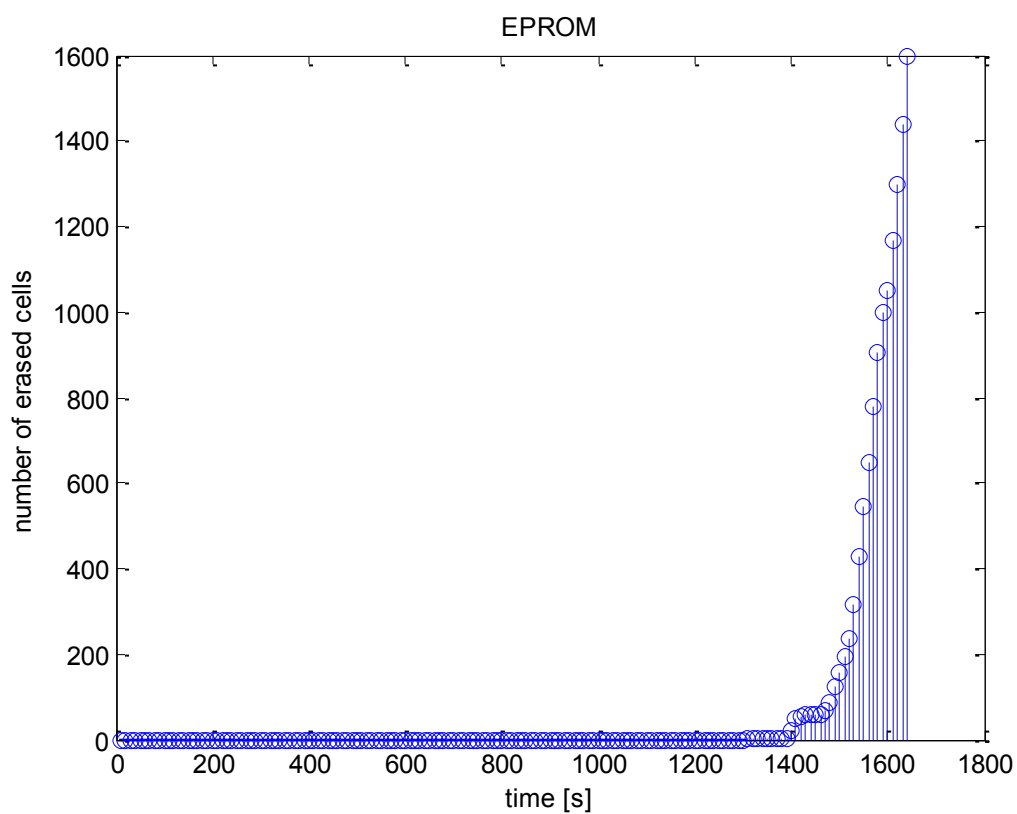
**Obrázek 72** Detail desky FPGA po ukončení měření, displej indikuje počet načtených záznamů.



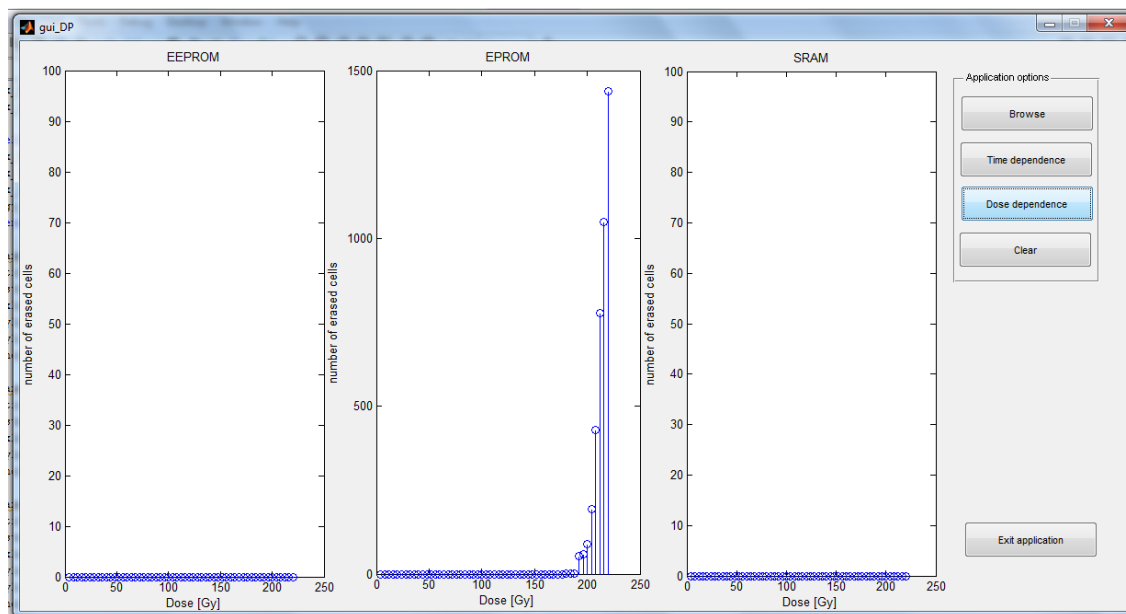
**Obrázek 70** Kontrola obsahu EEPROM, záznam z ChipScope.



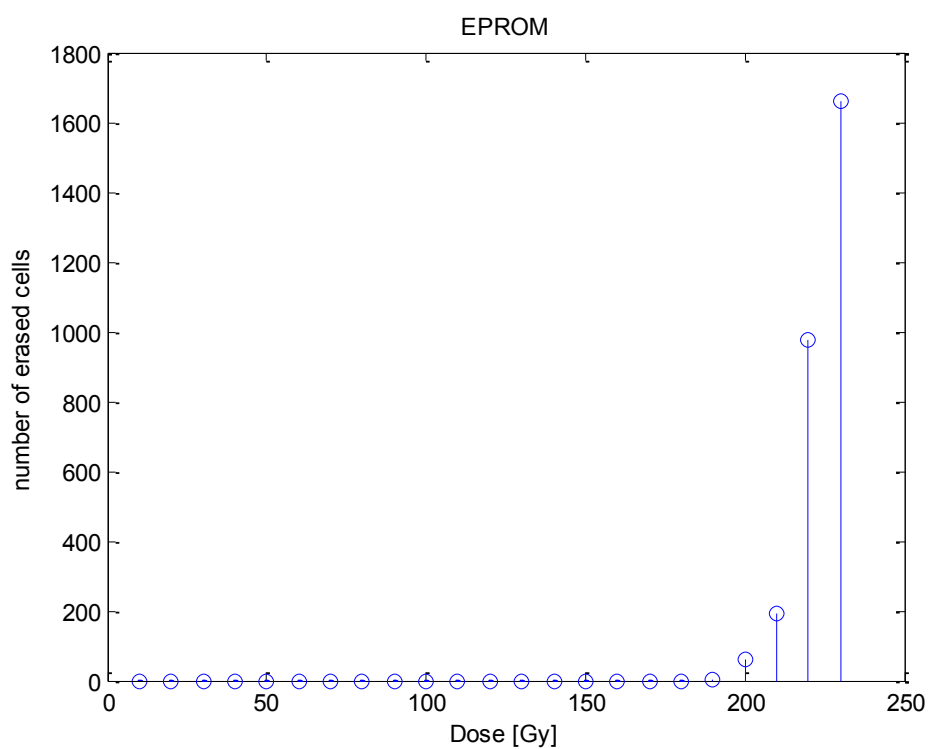
**Obrázek 74** Výsledek měření pro všechny paměti, počet smazaných buněk v závislosti na čase.



**Obrázek 73** Výsledek měření pro EPROM.



**Obrázek 75** Výsledek měření pro všechny paměti, počet smazaných buněk v závislosti na dávce.



**Obrázek 76** Výsledek měření pro EPROM

## 13. Zhodnocení dosažených výsledků

První úkolem práce bylo seznámení se s ionizujícím zářením používaným v biomedicině a jeho vlivem na polovodičové paměti. Vzhledem k tomu, že zatím není publikováno mnoho odborné literatury, zabývající se tímto problémem, byly zdrojem informací především články z vědeckých konferencí. Záření může mít na polovodiče účinky přechodné či trvalé. Přechodné účinky znamenají chybné signály (šum), případně chybné logické stavy. Destruktivní změny nebo těžce odstranitelné změny v polovodičové součástce jsou účinky trvalými a jejich náprava většinou vyžaduje odstavení součástky. Účinky ionizujícího záření jsou také vázané na množství radiační energie vyzářené na jednotku hmotnosti a na množství záření absorbované za jednotku času. Roli pak také hraje typ ionizujícího záření (gama fotony, neutrony, ...) nebo výrobní technologie dané polovodičové součástky.

Praktickou část práce tvoří návrh a realizace DPS přípravku pro průběžnou analýzu vybraných typů pamětí a návrh obvodové logiky FPGA přípravku. Při návrhu DPS se vycházelo z možností desky NEXYS3, která má omezený počet konektorů pro připojení logických signálů a z katalogových listů jednotlivých součástek. Návrh obvodové logiky přípravku prošel několika stádii. Od počátečního návrhu v jazyce VHDL přes testování všech modulů pomocí simulátoru až k ověření funkčnosti návrhu pomocí logického analyzátoru.

Měření vlivu ionizujícího záření na polovodičové paměti bylo realizováno na kobaltovém ozařovači TERAGAM a na přístroji CyberKnife. Analyzovány byly tři druhy pamětí: SRAM, EPROM, EEPROM. Vliv záření se nepotvrdil pro nízké dávky záření (desítky Gy), proto se na ozařovači nastavila velmi vysoká dávka 2000 Gy, která byla postupně během 30 min přípravkem absorbována, a sledovalo se, při jaké hodnotě se v pamětech začnou objevovat změny. Hraniční dávkou bylo přibližně 165 Gy. Nejdříve počet změn narůstal pomalu a od dávky 198 Gy byla frekvence výskytu změn vyšší. Při analýze výsledků v počítači bylo zjištěno, že změny nastaly pouze v obsahu paměti EPROM. SRAM a EEPROM zůstaly nezměněny. Lze tedy konstatovat, že paměti typu EPROM jsou ionizujícím zářením ovlivnitelné. Jednou z příčin nulových výsledků u SRAM a EEPROM může být pouzdro součástky. U EPROM je čip umístěn pouze pod sklíčkem, takže je přímo vystaven záření, naopak u pamětí EEPROM a SRAM je čip kryt pouzdem, jehož materiál může záření odstínit. V této práci byly analyzovány konkrétní typy pamětí, není tedy vyloučeno, že součástky jiných výrobců nebo s jiným pouzdem záření se budou chovat stejně jako v našem případě.

Návrhem obvodové logiky přípravku bylo využito jen malé procento jednotlivých součástí čipu FPGA (klopné obvody, LUT) z čehož vyplývá, že je možné návrh dále rozšířit. Např. motivací k další práci by mohla být vizualizace probíhajících změn v pamětech, v reálném čase na monitoru, protože jednou z dalších využitelných periférií desky NEXYS3 je VGA výstup.

## 14. Použitá literatura

- [1] ULLMANN, V. *Aplikace ionizujícího záření v radiologických oborech*. 1. vydání. Ostrava: Ostravská univerzita v Ostravě. 2013. 182 s. ISBN 978-80-7464-211-1.
- [2] ULLMANN, V. *Jaderná a radiační fyzika*. [online]. [cit.2014-01-05] URL: <<http://astronuklfyzika.cz/strana2.htm>>
- [3] CHVOSTKOVÁ, Z. *Přehled zobrazovacích metod ve zdravotnictví s podporou FPGA*. Ostrava: VŠB – Technická univerzita Ostrava. Fakulta elektrotechniky a informatiky. 2012, 87s. Vedoucí bakalářské práce Ing. Vladimír Kašík Ph.D.
- [4] ŠŤASTNÝ, J. *FPGA prakticky: Realizace číslicových systémů pro programovatelná hradlová pole*. 1. vydání. Praha: BEN – technická literatura. 2010. 200 s. ISBN 978-80-7300-261-9.
- [5] KAŠÍK, V. *Programovatelná hradlová pole*. 1. vydání. Ostrava: VŠB – Technická univerzita Ostrava. Fakulta elektrotechniky a informatiky. Katedra kybernetiky a biomedicínského inženýrství. 2012. 59s. Určeno pro projekt: Operační program Vzdělávání pro konkurenceschopnost.
- [6] HEINKEL, U. PADEFFKE, M. HAAS, W. BUERNER, T. BRAISZ, H. GENTNER, T. GRASSMANN, A. *The VHDL Reference: A practical Guide to Computer – Aided Integrated Circuit Design*. 1. Vydání. England: John Wiley & Sons, Ltd. 2000. 420s. ISBN 0 47189972 0.
- [7] *Unit 2: FPGA Architecture*. [online]. [cit.20014-09-25] URL: <[http://www.ami.ac.uk/courses/ami4460\\_fpga/u02/](http://www.ami.ac.uk/courses/ami4460_fpga/u02/)>
- [8] *Couplnig and Decoupling*. [online]. [cit.2014-08-16] URL: <<http://www.capacitorguide.com/coupling-and-decoupling/>>
- [9] *UM6114*. [online]. UMC.6s. URL: <<http://www.datasheetarchive.com/dl/Scans-067/DSA2IH00218667.pdf>>
- [10] *M27C64A*. [online]. SGH – Thompson Microelectronic. 11s. URL:< <http://www.alldatasheet.com/datasheet-pdf/pdf/22828/STMICROELECTRONICS/M27C64A.html>>
- [11] *25AA640/25LC640*. [online]. Arizona: Microchip, USA. 24s. [cit.20014-02-01]. URL:< <http://ww1.microchip.com/downloads/en/DeviceDoc/21223H.pdf>>
- [12] *Spartan-6 Family Overview*. [online]. California: Xilinx, Inc. 11s. [cit.2014-10-25]. URL:< [http://www.xilinx.com/support/documentation/data\\_sheets/ds160.pdf](http://www.xilinx.com/support/documentation/data_sheets/ds160.pdf)>
- [13] ANTOŠOVÁ, M. DAVÍDEK, V. *Číslicová technika*. 3. vydání České Budějovice: Knopp. 2003. 282 s. ISBN 978-80-7232-333-3
- [14] VALÁŠEK, P. LOSKOT, R. *Polovodičové paměti*. 1. Vydání. Praha: BEN – technická literatura. 1997 – 1998. 232 s. ISBN 80-86056-18-X.
- [15] *Async/Page/Burst CellularRAMTM 1.5*. [online]. Boise: Micron Technology Inc. 68s. [cit.2014-04-03]. URL:< <http://www.micron.com/parts/psram/cellularram/mt45w8mw16bgx-701-it#sim>>

- [16] *Ionizace*. [online]. [cit.2014-01-02].  
URL:< <http://cs.wikipedia.org/wiki/Ionizace>>
- [17] FETAHOVIĆ, I. PEJOVIĆ, M. VUJISIĆ, M. *Radiation Damage in Electronic Memory Devices*. [online]. [cit. 2013-05-25]. URL:  
< <http://www.hindawi.com/journals/ijp/2013/170269/>>.
- [18] CHUGG, M. A. *Ionising radiation effects: a vital issue for semiconductor electronics*. [online]. [cit. 2014-08-06]. URL:  
<<http://ieeexplore.ieee.org/xpl/abstractKeywords.jsp?arnumber=291651>>.
- [19] *Radiation Hardening*. [online]. [cit. 2015-02-27]. URL:  
<[http://en.wikipedia.org/wiki/Radiation\\_hardening#Digital\\_damage:\\_SEE](http://en.wikipedia.org/wiki/Radiation_hardening#Digital_damage:_SEE)>
- [20] *Photocurrent*. [online]. [cit. 2014-11-30]. URL:  
<<http://en.wikipedia.org/wiki/Photocurrent>>
- [21] *Spartan-6 FPGA Clocking Resources*. [online]. California: Xilinx, Inc. 116s. [cit.2013-12-20]. URL:<  
[http://www.xilinx.com/support/documentation/user\\_guides/ug382.pdf](http://www.xilinx.com/support/documentation/user_guides/ug382.pdf)>
- [22] *Absorbovaná dávka*. [online]. [cit. 2015-04-14]. URL:<  
[http://cs.wikipedia.org/wiki/Absorbovan%C3%A1\\_d%C3%A1vka](http://cs.wikipedia.org/wiki/Absorbovan%C3%A1_d%C3%A1vka)>
- [23] *Using Nonvolatile Static RAMs*. [online]. [cit. 2015-04-14]. URL:<  
<http://www.maximintegrated.com/en/app-notes/index.mvp/id/540>>
- [24] *Latchup*. [online]. [cit. 2015-04-14]. URL:<  
<http://en.wikipedia.org/wiki/Latchup>>
- [25] KAŠÍK, V. *Programování hradlových polí – Způsoby návrhu FPGA a CPLD*. 1. Vydání. Ostrava: VŠB – Technická univerzita Ostrava. 2012. 22 s.
- [26] *Ozařovače Teragam®*. [online]. [cit. 2015-04-14]. URL:<  
<http://isotrend.cz/produkty-a-sluzby/zdravotnictvi/ozarovace-teragam/>>
- [27] *Radioterapie*. [online]. [cit. 2015-04-14]. URL:<  
<http://astronuklfyzika.cz/JadRadMetody.htm#TeleTerapie>>
- [28] *AVI Medical*. [online]. [cit. 2015-04-14]. URL:< <http://avi-medical.com/partners/apparatus/>>
- [29] *Calculation of Monitor Units*. [online]. [cit. 2015-04-14]. URL:<  
<http://ozradonc.wikidot.com/calculation-of-monitor-units-treatment-time>>
- [30] *Dual Inline Packages (DIPs) and Memory Modules*. [online]. [cit. 2015-05-01]. URL:< <http://www.pcguides.com/ref/ram/packDIP-c.html>>
- [31] *Small Outline Integrated Circuit*. [online]. [cit. 2015-05-01]. URL:< [http://en.wikipedia.org/wiki/Small\\_Outline\\_Integrated\\_Circuit](http://en.wikipedia.org/wiki/Small_Outline_Integrated_Circuit)>
- [32] *Vnitřní paměti*. [online]. [cit. 2015-05-01]. URL:< <http://www.fj.muni.cz/usr/pelikan/ARCHIT/TEXTY/INTPAM.HTML>>



## 15. Přílohy

- I. Data
- II. DP2 – ISE project
- III. Gui\_DP – Matlab project
- IV. MemtestTerminal – Microsoft Visual studio 2010 project
- V. Schema – Eagle project
- VI. Statavové diagramy – Enterprise Architect project

### **Obsah CD**

Diplomova prace.pdf

Prilohy.zip